



Modèles et mécanismes d'adaptation de l'interaction Homme-Machine aux changements de contexte

Thomas Villaren

► To cite this version:

Thomas Villaren. Modèles et mécanismes d'adaptation de l'interaction Homme-Machine aux changements de contexte. Interface homme-machine [cs.HC]. Télécom Bretagne, Université de Bretagne-Sud, 2012. Français. NNT : . tel-00789551v2

HAL Id: tel-00789551

<https://theses.hal.science/tel-00789551v2>

Submitted on 15 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THESE / Télécom Bretagne

sous le sceau de l'Université européenne de Bretagne

pour obtenir le grade de Docteur de Télécom Bretagne

En habilitation conjointe avec l'Université de Bretagne-Sud

École doctorale Sicma

mention : Sciences et technologies de l'information et de la communication

présentée par

Thomas Villaren

préparée dans le département Logique des usages,
sciences sociales et de l'information

Laboratoire Labsticc

Modèles et mécanismes d'adaptation de l'interaction homme-machine aux changements de contexte

Thèse soutenue le 30 novembre 2012
Devant le jury composé de :

Philippe Palanque

Professeur, Université Paul Sabatier - Toulouse / président

Christophe Kolski

Professeur, Université de Valenciennes / rapporteur

Dominique Scapin

Directeur de recherche, Inria - Le Chesnay / rapporteur

Franck Poirier

Professeur, Université de Bretagne-Sud / examinateur

Angélica Leal

Ingénieur, Bertin Technologies - St Quentin en Yvelines / examinatrice

François Legras

Ingénieur, Deev Interaction - Plouzané / examinateur

Gilles Coppin

Professeur, Télécom Bretagne / directeur de thèse

Didier Bazalgette

Ingénieur, Direction Général de l'Armement - Bagneux / invité

Dominique Soler

Directeur, Bertin Technologies - St Quentin en Yvelines / invité



Sous le sceau de l'Université européenne de Bretagne

Télécom Bretagne

En habilitation conjointe avec l'Université de Bretagne Sud

École Doctorale – SICMA

Modèles et mécanismes d'adaptation d'Interaction Homme-Machine aux changements de contexte

Thèse de Doctorat

Mention : « Sciences et Technologies de l'Information et de la Communication »

Présentée par **Thomas Villaren**

Département : Logique des Usages, Sciences Sociales et de l'Information (LUSSI)

Laboratoire : LabSTICC Pôle : CID

Directeur de thèse : Prof. Gilles Coppin

Soutenue le 30 novembre 2012

Jury :

Président :	M. Philippe Palanque, Professeur, Université Paul Sabatier / Toulouse 3
Rapporteurs :	M. Christophe Kolski, Professeur, Université de Valenciennes M. Dominique Scapin, Directeur de Recherche, INRIA
Examineurs :	M. Franck Poirier, Professeur, Université de Bretagne Sud Mme Angélica Leal, Consultant, Bertin Technologies (encadrant) M. François Legras, Président, Deev Interaction (encadrant)
Invités :	M. Didier Bazalgette, Ingénieur, Direction Générale de l'Armement M. Dominique Soler, Directeur, Bertin Technologies

Remerciements

Trois ans. Trois ans résumés en quelques 300 pages. Trois ans partagés entre la mer et la ville, à régulièrement avaler les kilomètres du Paris-Brest, du milieu académique au milieu industriel, et inversement. Deux mondes et trois années qui m'ont permis de mener à bien ces travaux, ce qui n'aurait pas été possible sans celles et ceux que je souhaiterais remercier ici.

Mes premiers remerciements vont à Angélica Leal, sans qui je n'aurais probablement jamais terminé cette thèse dans le temps imparti. Merci Angélica de m'avoir laissé autonome tout en me guidant dans le monde du conseil et, plus important encore, en accompagnant mes premiers pas dans celui de l'ergonomie.

Merci également à Gilles Coppin, mon directeur de thèse, qui malgré un emploi du temps (sur-)chargé, a toujours su se dégager un créneau lors de mes excursions brestoises (certes, parfois après avoir dû « camper » devant son bureau pendant quelques heures). La richesse des échanges que nous avons alors suffisait à contrebalancer leur petit nombre.

Merci aussi à François Legras, de m'avoir encadré et d'avoir répondu à mes questions souvent dans des délais très courts, en dépit de son agenda également très rempli.

Un grand merci aux membres de mon jury qui m'ont fait l'honneur d'accepter de lire et évaluer mes travaux : Christophe Kolski et M. Dominique Scapin, qui ont eu à lire les nombreuses pages de ce manuscrit et en rédiger un rapport de lecture dans un temps réduit, ainsi que Philippe Palanque et Franck Poirier, qui ont examiné mes travaux avec attention. Merci à tous les quatre pour l'intérêt porté à mes travaux.

Je souhaite remercier M. Dominique Soler, directeur de l'activité Ergonomie et Facteurs Humains de Bertin Technologies, qui m'a permis de réaliser cette thèse en convention CIFRE et qui m'a fait confiance lorsque j'intervenais sur projets, ainsi que Didier Bazalgette, responsable du domaine Homme & Systèmes à la Mission pour la Recherche et l'Innovation Scientifique (MRIS) de la Direction Générale de l'Armement, pour m'avoir suivi au cours de ces travaux et apporté ses conseils avisés. Je remercie également la DGA qui a participé au financement de cette thèse à travers l'allocation d'une bourse CIFRE-Défense.

Je remercie ensuite les deux équipes au sein desquelles j'ai évolué au cours de ces trois années de thèse. Du côté parisien, l'équipe « ergo » de Bertin Technologies, que j'ai tenté de convertir au côté obscur de la force (celui des informaticiens, bien sûr)

mais en vain, et qui à l'inverse, m'a converti à l'ergonomie et grâce à qui j'ai beaucoup appris. Merci Anaïs, Camille, Caroline, Charlie, Christelle, Julien, Mariana, Marie, Nathalie, Nicolas, Robin, Sandrine, Stéphanie, Thibaud et Thomas.

Du côté de Brest, je remercie toute l'équipe du département LUSI, en particulier celles et ceux qui m'ont accueilli dans leurs bureaux, me trouvant un coin de table pour travailler lors de mes passages à l'École : Benoit, Cyril, Emile, Emilien, Ikbel, Mathieu, Romain, Sahbi, Santiago, Yannick et Ghislaine.

Merci à ma famille, en particulier mes parents Marc & Martine, et mon frère, Tanguy, qui me soutiennent (et me supportent !) depuis plus de 26 ans : si je suis là aujourd'hui, c'est en grande partie grâce à vous.

Un grand merci aussi à tous mes amis, qui m'ont soutenu tout au long de mes travaux (« alors, ça avance la thèse ? ») et qui supportent mon humour depuis plus de 10 ans pour certains. Merci Cynthia, Hélène, Matt et Stouph, les bretons : Adry, Agnès, Antho, Aude, Didou, Elodie, Greg, Gwladys, Ju, KPE, Loïc, Mél', Nono, PY, les Teubreux (ou presque) : Chloé, Clara, Fred, Greg, Guillaume & Guillaume, Gulzhan, Jérém', Jeff, Lucile, MS, Nadia, Poule, Romain, Sandra, Simon, Sophie, Steph, Tus, Vince et les autres. Un merci particulier à mes relecteurs, à savoir dans l'ordre des chapitres : Marc, Fred, M & M's, Matt, Martine, Guillaume, Cynthia, LN, El Commandante, Pixi, Sred.

Et merci à tous ceux que j'ai rencontré, croisé et avec qui j'ai échangé au cours de ces années et dont le nom n'apparaît pas ici, je pense à vous également.

Merci – Thank you – Trugarez – Obrigado – Grazie

Gracias – Chokrane – Rahmet – Mulțumesc

Table des matières

1	Introduction générale	1
1.1	Approche	2
1.1.1	Domaine d'étude	3
1.1.2	Cadre de l'étude	3
1.2	Structure de la thèse	4
I	Transitions & Contextes : une revue de la littérature	7
2	Modéliser les transitions	9
2.1	Les modèles de tâches	9
2.1.1	Les différentes approches de modélisations de tâches	10
2.1.2	L'intégration des transitions dans les modèles de tâches	11
2.1.3	Comparaison des notations	16
2.1.4	Commentaire sur les modèles de tâches	18
2.2	Le Task Switching, ou l'approche par le « coût cognitif »	18
2.2.1	Préambule	19
2.2.2	Les interprétations du « coût de changement »	21
2.2.3	Commentaire sur le Task Switching	25
2.3	Les interruptions en Interaction Homme-Machine	25
2.3.1	Taxinomie de l'interruption humaine	25
2.3.2	La psychologie de l'interruption	28
2.3.3	Coordination de l'interruption : comment interrompre et à quel moment ?	34
2.3.4	Les interruptions dans les modèles	42
2.3.5	Commentaire sur les Interruptions	43
2.4	Synthèse du chapitre 2	44
3	Modéliser le contexte des transitions	47
3.1	La Situation Awareness, un concept aux définitions multiples	48
3.2	Les modèles de Situation Awareness	50
3.2.1	Modèles descriptifs de SA	50
3.2.2	Modèles prescriptifs de SA	55
3.2.3	Synthèse sur les modèles de SA individuelle	61
3.3	La Situation Awareness dans les systèmes collaboratifs	62
3.3.1	La SA en équipe	63
3.3.2	La SA distribuée	65
3.4	Mesurer la Situation Awareness	69

3.4.1	Revue des techniques de mesure de la Situation Awareness . . .	69
3.4.2	Quelques méthodes de mesure de la SA	72
3.4.3	Commentaires concernant la mesure de la SA	75
3.5	Synthèse du chapitre 3	76

II De la modélisation à l'analyse des transitions, contribution conceptuelle 79

4	Les transitions et la dualité tâche / contexte	81
4.1	Une proposition de taxinomie des transitions	82
4.1.1	La logique des intervalles temporels de Allen	82
4.1.2	Les deux dimensions de la taxinomie	84
4.1.3	Description de la taxinomie	86
4.2	Vers un nouvel espace de modélisation des transitions	94
4.2.1	Préambule : le modèle DSA	94
4.2.2	Critiques et adaptation du modèle DSA	95
4.2.3	Choix du modèle de tâches	97
4.2.4	Synthèse de l'approche duale	101
4.3	Analyse des transitions : prédiction et aide à la conception	102
4.3.1	Réflexions sur l'analyse prédictive des scénarios	103
4.3.2	Synthèse des réflexions	112
5	La méthodologie SAMANTA	113
5.1	Présentation de la méthodologie	114
5.2	Construction des modèles	116
5.2.1	Extraction des connaissances	116
5.2.2	Construction du modèle de tâches	117
5.2.3	Construction du modèle de SA	128
5.2.4	Définition des contextes associés aux tâches	133
5.2.5	Synthèse de la première phase	137
5.3	Analyse des transitions	138
5.3.1	Préambule : les transitions dans le modèle de tâches	139
5.3.2	Statistiques générales sur la transition	140
5.3.3	Recherche de sous-graphes communs	144
5.3.4	Analyse des éléments saillants	148
5.3.5	Analyse des métadonnées associées à la transition	150
5.3.6	Intégration des tâches parallèles dans l'analyse	152
5.4	Impact des transitions sur l'IHM	155
5.4.1	Les normes et guides en ergonomie et facteurs humains	156
5.4.2	Intégration des normes et guides à la méthodologie	161
5.4.3	Intégration à une approche générative des interfaces	163
5.5	Illustration de la méthodologie : application à un jeu vidéo	164

5.5.1	Introduction	164
5.5.2	Modèle de tâches	166
5.5.3	Définition des contextes associés aux tâches	166
5.5.4	Analyse des transitions	169
5.5.5	Synthèse du cas applicatif	175
5.6	Synthèse du chapitre 5	176

III Mise en œuvre de l'approche, contribution pratique177

6	Outillage de la méthodologie SAMANTA	179
6.1	Démarche	179
6.2	Champ d'application de l'outil développé	180
6.3	Spécifications techniques de l'outil	181
6.3.1	Choix de la technologie	181
6.3.2	Données d'entrée de l'outil	185
6.3.3	Architecture de l'application	187
6.4	Description de l'outil	192
6.4.1	Zones de l'écran	192
6.4.2	Fonctions de l'outil	193
6.5	Perspectives d'évolution de l'outil	203
6.5.1	Mise en place d'autres méthodes d'analyse des transitions . . .	203
6.5.2	Identification des transitions difficiles	203
6.5.3	Intégration d'autres outils à la méthodologie	204
6.6	Synthèse du chapitre 6	205
7	Opérationnalisation de la méthodologie	207
7.1	Les systèmes de drone	207
7.2	Mise en œuvre de la méthodologie SAMANTA dans les systèmes de drone	209
7.3	Extraction des connaissances	210
7.3.1	Sources de données	210
7.3.2	Exploitation des données collectées	211
7.4	Modélisation	211
7.4.1	Modèle de tâches	212
7.4.2	Modèle de Situation Awareness	214
7.5	Définition et validation opérationnelle des contextes	222
7.6	Analyse de transitions : choix du scénario et des transitions	224
7.6.1	Description du scénario	224
7.6.2	Formalisation du scénario	225
7.6.3	Analyse de transition #1 : changement d'optique en contrôle manuel de la charge utile	227

7.6.4	Analyse de transition #2 : changement de mode de contrôle de la charge utile	236
7.7	Synthèse de l'analyse du cas applicatif	241
7.8	Autre application de la méthodologie SAMANTA	242
7.8.1	Présentation du cas d'étude	242
7.8.2	Démarche adoptée	243
7.8.3	Synthèse des commentaires	243
7.8.4	Synthèse des recommandations de l'expert	244
8	Conclusion générale	247
8.1	Résumé des contributions	247
8.1.1	Contributions conceptuelles	248
8.1.2	Contributions pratiques	248
8.2	Perspectives	249
8.2.1	Perspectives concernant la taxinomie	249
8.2.2	Perspectives concernant l'outil	249
8.2.3	Perspectives concernant la méthodologie	250
	Publications	255
	Bibliographie	256
	Annexes	275
A	Complément bibliographique	277
B	Description fonctionnelle de l'outil SAMANTA	285
C	Détails techniques sur l'outil SAMANTA	291
D	Le PEA FH/PA	305
E	Compléments sur l'application aux systèmes de drones	309

Table des figures

2.1	expressivité versus complexité pour les modèles de tâches	10
2.2	exemple de tâche de Stroop	19
2.3	la décomposition des processus exécutifs lors d'un changement de tâche	24
2.4	le modèle de gestion des interruptions de Latorella	29
2.5	chronologie d'une interruption (Trafton et Monk, 2007)	30
2.6	exemples d'indications pour aider la reprise d'interruption	32
2.7	propositions de solutions d'interruption sur un plan coût/charge . . .	34
2.8	captures d'écran de l'expérimentation de McFarlane	35
2.9	taxinomie des interruptions proposées par Gievska et Sibert (2004). .	38
2.10	dépendances des variables de la taxinomie et influence sur le timing de l'interruption	39
2.11	phases du modèle d'apprentissage de Godbole et Smarie	40
2.12	hiérarchie de décomposition des tâches de Adamczyk et Bailey	41
2.13	intégration du framework OASIS	41
3.1	modèle à 3 niveaux de Endsley (1995b) (adaptation).	51
3.2	modèle de Endsley <i>et al.</i> (2000a) appliqué au fantassin (adaptation). .	53
3.3	les 3 modes d'évaluation du modèle <i>Recognition Primed Decision</i> . . .	54
3.4	modèle opérateur du logiciel MIDAS (extrait de Smith et Tyler (1997)).	56
3.5	modèle SAMPLE de Zacharias <i>et al.</i> (1996).	58
3.6	la représentation du jugement selon Brunswik (1956)	59
3.7	<i>Expanded Lens Model</i> de Stewart et Lusk (1994)	60
3.8	schématisation de la SA d'une équipe de 3 membres	64
3.9	modèle de la SA par équipe résumé par Salmon <i>et al.</i> (2008a).	64
3.10	réseau propositionnel dans le domaine de la distribution d'énergie . .	67
4.1	cinq relations ensemblistes de la dimension contextuelle	86
4.2	transition simple – Séparation des contextes.	87
4.3	transition simple – Intersection des contextes.	88
4.4	transition simple – Réduction de contexte.	88
4.5	transition simple – Extension de contexte.	88
4.6	transition simple – Redondance totale des contextes.	89
4.7	ajout de tâche – Séparation des contextes	89
4.8	ajout de tâche – Intersection des contextes	90
4.9	ajout de tâche – Réduction de contexte	90
4.10	ajout de tâche – Extension de contexte	90
4.11	ajout de tâche – Redondance totale des contextes	91
4.12	interruption de tâche – Séparation des contextes	91

TABLE DES FIGURES

4.13	interruption de tâche – Intersection des contextes	92
4.14	interruption de tâche – Réduction de contexte	92
4.15	interruption de tâche – Extension de contexte	93
4.16	interruption de tâche – Redondance totale des contextes	93
4.17	DSA : exemple d'association	95
4.18	transposition du paradigme multi-opérateur / mono-tâche au cas mono- opérateur / multitâche	98
4.19	principe d'association entre les éléments de situations et l'arbre CTT.	102
4.20	exemple d'arbre de tâche	104
4.21	description des tâches d'un contrôleur aérien militaire	106
4.22	description du scénario de contrôle aérien militaire	107
5.1	description générale de la méthodologie SAMANTA	116
5.2	sélection de l'identificateur temporel dans l'éditeur e-COMM	119
5.3	illustration du principe de factorisation des tâches	121
5.4	représentation des paramètres des tâches dans l'éditeur CTTE	123
5.5	principe de notation des nœuds d'un arbre enraciné.	127
5.6	création d'un réseau propositionnel à partir d'entretiens	130
5.7	illustration de la notation associée aux réseaux propositionnels.	132
5.8	association des éléments de situation à une tâche feuille.	134
5.9	illustration de l'association tâche / contexte	136
5.10	transition entre 2 tâches non voisines avec le même ascendant direct.	140
5.11	transition entre 2 tâches non voisines n'ayant pas le même parent.	141
5.12	un exemple de transition entre deux contextes.	141
5.13	illustration de l'application de l'algorithme LCSG.	147
5.14	visualisation de la saillance d'un élément	149
5.15	différence de modes sur le McDonnell Douglas MD-82	150
5.16	exemple de transition visuelle, animation du paramètre de vitesse.	152
5.17	recherche de tâches parallèles	153
5.18	recherche des tâches exécutées en parallèle d'une transition	154
5.19	utilisation des tags afin d'assurer la cohérence de l'interface	162
5.20	arbre CTT décrivant les tâches d'un joueur de Battlefield 3.	167
5.21	graphe de <i>Situation Awareness</i> pour le joueur de Battlefield 3.	168
5.22	captures d'écran d'une partie de Battlefield 3	170
5.23	schématisation de la transition analysée.	171
5.24	la transition considérée vis-à-vis de la taxinomie	172
5.25	les 3 plus grands sous-graphes communs de la transition	174
6.1	les étapes 2 à 4 de la méthodologie et les outils associés.	181
6.2	les technologies soutenant les étapes 2, 3 et 4 de la méthodologie	187
6.3	diagramme UML de classe de représentation de l'arbre CTT.	188
6.4	diagramme UML de classe du graphe de situations.	189
6.5	les modèles manipulés par l'outil.	191

TABLE DES FIGURES

6.6	photo de l'utilisation de l'outil sur une tablette tactile iPad.	192
6.7	capture d'écran des trois zones principales de l'outil SAMANTA.	193
6.8	capture d'écran des options d'importation de données.	194
6.9	capture d'écran de la vue « arbre de tâches ».	195
6.10	capture d'écran de la miniature lors d'un survol souris	195
6.11	capture d'écran de la fenêtre modale de détail d'une tâche.	197
6.12	capture d'écran de la vue « graphe de <i>Situation Awareness</i> ».	197
6.13	capture d'écran du menu déroulant de sélection de la tâche courante .	198
6.14	capture d'écran de la fenêtre modale de détails d'un élément	199
6.15	capture d'écran de la vue centrale de l'onglet « Analyse ».	199
6.16	capture d'écran de la fenêtre modale de sélection d'une transition. . .	200
6.17	capture d'écran – fenêtre modale d'analyse d'une transition, onglet 1	201
6.18	capture d'écran – fenêtre modale d'analyse d'une transition, onglet 2	201
6.19	capture d'écran – fenêtre modale d'analyse d'une transition, onglet 3	202
7.1	le drone SDTI construit par Sagem DS	208
7.2	le drone Harfang construit par Cassidian AS.	209
7.3	vue d'ensemble du modèle de tâches.	212
7.4	la tâche « Gérer l'environnement » et ses sous-tâches.	213
7.5	la tâche « Piloter le drone » et ses sous-tâches.	213
7.6	la tâche « Gérer la Charge Utile » et ses sous-tâches.	215
7.7	les six groupes du graphe global de <i>Situation Awareness</i>	217
7.8	les éléments du graphe global relatifs au drone.	218
7.9	les éléments du graphe global relatifs à la météorologie.	218
7.10	les éléments du graphe global relatifs à la charge utile du drone. . . .	219
7.11	les éléments du graphe global relatifs à transmission des informations	220
7.12	les éléments du graphe global relatifs aux pistes détectées / recherchées.	221
7.13	les éléments du graphe global relatifs à l'environnement.	221
7.14	aperçu des contextes définis pour quelques tâches du scénario.	223
7.15	le modèle de tâches et les étiquettes formalisées associées.	225
7.16	représentation des états du scénario et des tâches associées.	226
7.17	représentation des états du scénario	227
7.18	première transition vis-à-vis de la taxinomie	229
7.19	illustration d'un changement d'optique	230
7.20	proposition d'aide à la transition	231
7.21	zones DRI et historique de la trace senseur	232
7.22	sous-graphe associé aux éléments de contrôle manuel de la charge utile.	233
7.23	sous-graphe associé aux différents aspects des pistes.	235
7.24	indication sur la vue caméra de la direction des deux dernières pistes	236
7.25	la deuxième transition vis-à-vis de la taxinomie	237
7.26	partie du sous-graphe commun relatif à l'optique infrarouge.	240
7.27	partie du sous-graphe commun relative à la piste suivie.	241

TABLE DES FIGURES

8.1	illustration de l'évolution de l'activation des éléments de situation . . .	252
A.1	les cinq étapes de la méthode <i>Critical Decision Method</i>	279
E.1	la tâche « Gérer l'environnement » et ses sous-tâches.	309
E.2	contexte associé à la tâche « Gérer la météo ».	310
E.3	contexte associé à la tâche « S'orienter dans l'environnement ».	310
E.4	la tâche « Piloter le drone » et ses sous-tâches.	311
E.5	contexte associé à la tâche « Suivi de plan de vol ».	311
E.6	contexte associé à la tâche « Orbite Automatique ».	312
E.7	contexte associé à la tâche « Fly The Footprint ».	313
E.8	contexte associé à la tâche « Centre orbite ».	313
E.9	tâche « Gérer la Charge Utile » et ses sous-tâches.	314
E.10	contexte associé à la tâche « Gérer les paramètres CU ».	315
E.11	contexte associé à la tâche « FixForward + E/O ».	315
E.12	contexte associé à la tâche « Rechercher en E/O ».	316
E.13	contexte associé à la tâche « Rechercher en IR ».	317
E.14	contexte associé à la tâche « Surveiller cible en IR ».	317

Liste des tableaux

2.1	comparaison de neuf notations de modélisation des tâches	17
2.2	recommandations sur le choix du mode de coordination	37
4.1	les relations de Allen (1983), nommées par Vernier et Nigay (2001) . .	83
4.2	synthèse des transitions décrites sur les axes temporel et contextuel. .	94
4.3	comparaison de neuf notations de description des tâches	100
5.1	les identificateurs temporels de CTT, adapté de Paternò (2003). . . .	120
5.2	correspondance entre les identificateurs CTT et les catégories de transitions de la taxinomie.	126
5.3	correspondance entre les identificateurs CTT et les opérations ensemblistes pour la combinaison de contextes.	137
5.4	résumé des principes de conception intégrant la SA, première partie .	157
5.5	résumé des principes de conception intégrant la SA, deuxième partie .	158
6.1	langages de programmation des applications natives sur ordinateur et supports mobiles	183
A.1	questions posées lors de l'entretien en utilisant la méthode des décisions critiques (traduit de Klein <i>et al.</i> , 1989).	282
C.1	table de compatibilité des navigateurs pour l'outil SAMANTA.	296



1

Introduction générale

Novembre 2010. Laboratoire Technico-Opérationnel de la DGA. Arcueil (94), France. Dialogue entre les deux membres d'équipage en charge de deux drones d'observation : l'opérateur *Vecteur* pilote les deux drones et l'opérateur *Charge Utile* s'occupe des deux caméras :

“

- Opérateur *Vecteur* pour opérateur *Charge Utile*, le drone V_1 est stabilisé sur zone, je fais une orbite autour de la cible pour début mission reconnaissance, confirmez.
- Opérateur *Charge Utile* pour opérateur *Vecteur*, bien reçu. Début de reconnaissance. Opérateur *Vecteur*, pouvez-vous ramener le drone V_2 sur sa zone ? Il part vers l'ouest là !

”

S'ensuit un monologue de l'opérateur *Vecteur*, après avoir cherché quelques secondes le drone V_2 sur son interface :

“

- Mais, qu'est-ce qu'il fait ce drone ? Il est où ? Pourquoi il s'en va ?
- ...
- Ah, il y avait une alarme de dérive due au vent.
- Mince, je l'avais pas vue.
- Bon, il faut que je le ramène sur zone ; heureusement qu'il était assez haut, j'aurais pu percuter le relief.

”

Fort heureusement, le drone en question n'était qu'une entité informatique implémentée au sein d'un simulateur. Mais eut-elle eu lieu en conditions réelles de vol, cette erreur aurait pu avoir des conséquences bien plus graves sur l'intégrité du matériel et des populations alentour...

L'extrait rapporté ici est issu des enregistrements de la deuxième campagne expérimentale d'un projet d'étude amont de la Direction Générale de l'Armement (présenté en annexe D). L'un des objectifs de ce projet était l'étude, du point de vue des facteurs humains, du contrôle conjoint de deux drones par un seul opérateur.

La volonté de déléguer la supervision de drones multiples à un unique opérateur se traduit, dans les travaux actuels, par l'augmentation de l'automatisation de ces systèmes de drones (Cummings *et al.*, 2012a). Cette tendance à l'automatisation n'est pas restreinte au domaine militaire mais se vérifie également dans les domaines industriels tel que l'aéronautique (Billings, 1997), le contrôle de trafic aérien (Kaber *et al.*, 2005) ou la conduite de centrale nucléaire (Labarthe *et al.*, 2011).

L'opérateur en charge de ces systèmes est amené à gérer simultanément un *ensemble de tâches évoluant dynamiquement* dans le temps, et imposant des *changements fréquents de la situation tactique* au cours d'une même mission. Ces changements de contexte informationnel sont provoqués par le passage d'une tâche à une autre (dans notre exemple, le contrôle de deux entités distinctes), mais aussi par l'ajout de nouvelles tâches, le retrait ou l'interruption de tâches courantes, et la reprise de tâches.

Ces transitions entre tâches peuvent induire une rupture de la conscience de la situation¹ (*Situation Awareness*) (Jones et Endsley, 1996; Dismukes *et al.*, 1998; Kaber *et al.*, 2005), diminuant leur performance et augmentant la charge cognitive nécessaire à la reconstruction d'une image mentale *correcte* de la situation tactique (Amalberti, 1996).

1.1 APPROCHE

Nous proposons dans ce mémoire d'étudier l'impact des transitions sur la *Situation Awareness* des opérateurs en charge de systèmes dynamiques complexes. Nous étudions également la façon dont ce concept de « changement de *contexte*² » peut être intégré en phase de conception de nouveaux systèmes, afin de proposer des interfaces adaptées à ces situations changeantes et aux ruptures qu'elles engendrent.

La problématique sera abordée à travers deux questions :

Où ont lieu ces ruptures de *Situation Awareness*? C'est-à-dire, quels sont les changements de tâches qui nécessitent cette adaptation de l'image mentale?

Nous tâcherons de répondre à cette question en identifiant quand ces ruptures peuvent avoir lieu, en particulier par la mise en exergue, en amont de la conception, de celles qui pourraient avoir un impact sur l'opérateur.

Comment ces ruptures impactent-elles l'opérateur et sa *Situation Awareness*? Ce deuxième axe d'étude permet d'identifier les impacts qu'ont ces changements

1. Définie par Endsley (1995b) comme la capacité de l'opérateur à percevoir, comprendre et anticiper l'état de son environnement (chapitre 3).

2. Nous définissons le contexte d'une tâche comme l'ensemble des informations nécessaires à l'exécution nominale de cette tâche par l'opérateur.

de contexte sur l'opérateur et la manière dont l'interface peut être améliorée pour accompagner au mieux l'opérateur lors de ces transitions.

Afin d'étudier ces deux axes, le « où » et le « comment », nous adoptons deux orientations déterminantes :

- Nos travaux seront emprunts d'une *approche centrée sur les utilisateurs*, en intégrant le point de vue de l'utilisateur dès les prémices de la conception du système, et non uniquement ceux du concepteur ou de l'ergonome.
- Nous nous appuierons également sur la définition de modèles, qui nous permettront de formaliser la description des *tâches prescrites*³ de l'utilisateur et des connaissances nécessaires à la réalisation de ces tâches.

1.1.1 Domaine d'étude

Notons que les travaux présentés dans ce mémoire concernent les systèmes dynamiques complexes, par opposition aux systèmes grands publics (applicatif ou matériel). Ces systèmes dynamiques sont définis par Hoc (1996) comme des systèmes où « l'opérateur humain ne détermine pas entièrement les modifications de son environnement de travail ». La caractéristique de leur dynamique varie : par exemple, la gestion de processus industriels tels que la conduite de centrale nucléaire présente généralement une dynamique lente mais dans des domaines tels que l'aéronautique ou le contrôle de trafic aérien, les opérateurs humains doivent faire face à une forte dynamique temporelle et à des tâches changeant rapidement.

1.1.2 Cadre de l'étude

Enfin, cette thèse a été réalisée dans le cadre d'une Convention industrielle de formation par la recherche (CIFRE), établie entre Télécom Bretagne et l'entreprise Bertin Technologies, où le doctorant était intégré au pôle « Ergonomie des systèmes et Facteurs Humains » du département Bertin Conseil. L'activité du pôle Ergonomie couvre un large spectre des interventions « classiques » de l'ergonomie et des facteurs humains. En particulier, l'équipe d'ergonomes intervient en phase de conception et d'évaluation de nombreux systèmes complexes tels que la conception de salle de contrôle pour la conduite de centrale nucléaire, la conception, l'évaluation et la certification de cockpit pour l'aéronautique civile, le contrôle de trafic aérien, ou encore l'intégration des facteurs humains dans les programmes militaires...

L'intégration du doctorant en informatique au sein de cette équipe permet d'aborder la problématique identifiée en intégrant des composantes des domaines des facteurs humains et de l'informatique, en mettant notamment en place une méthodologie de conception répondant aux usages des ergonomes.

3. Leplat (1997) définit la *tâche prescrite* comme « le modèle de la tâche à réaliser, modèle que son concepteur destine à un opérateur. [Elle] est le fruit d'activités antérieures ».

1.2 STRUCTURE DE LA THÈSE

Cette thèse est composée de trois parties et de huit chapitres.

Le **chapitre 1** constitue l'introduction générale au sujet traité dans ce mémoire.

La **première partie** présente une revue de la littérature.

- Le **chapitre 2** introduit la notion de transition sous l'angle de la *dynamique temporelle*. Il présente les points de vue de l'Interaction Homme-Machine et de la Psychologie Cognitive sur le sujet.
- Le **chapitre 3** définit la notion de *Situation Awareness* et présente les approches de modélisation et de mesure relatives à ce concept. Les modèles présentés décrivent les processus d'acquisition et/ou de description de cette *Situation Awareness*, sur laquelle porte notre étude. Le but de ce chapitre est de déterminer comment l'image mentale de la situation tactique que construit l'opérateur peut être modélisée et comment la dynamique temporelle des transitions, étudiée au **chapitre 2**, a été intégrée à ces modèles.

La **deuxième partie** décrit les contributions conceptuelles de cette thèse.

- Devant le constat qu'il n'existe pas de modèle intégrant à la fois la dynamique temporelle des tâches et la description de l'évolution de situation tactique associée, le **chapitre 4** propose un nouveau modèle palliant ce manque. La définition de ce modèle s'appuie sur deux modèles décrits dans les **chapitres 2 et 3**.

Ce modèle *dual* soutient une taxinomie des transitions, introduite également au **chapitre 4**. Cette taxinomie établit une liste exhaustive des transitions prototypiques, caractérisées sur deux axes : temporel et contextuel.

Enfin, ce chapitre expose des solutions au premier axe de recherche, identifié par la question du « où ». Ces solutions cherchent à filtrer les transitions décrites dans un scénario « type » d'utilisation du système pour cibler, dans une analyse approfondie, les transitions jugées comme les plus « difficiles ».

- Le **chapitre 5** propose une méthodologie de conception, intitulée SAMANTA (*Situation Awareness Modeling and ANalysis for Transition Amelioration*), reposant sur une analyse de l'activité de l'opérateur et la description des tâches prescrites et des contextes associés pour évaluer « comment les transitions vont avoir un impact sur l'opérateur ». Cette méthodologie s'appuie sur le modèle dual introduit au **chapitre 4**.

Ce chapitre décrit les différentes méthodes d'analyse des transitions introduites dans la méthodologie SAMANTA et illustre cette analyse au travers d'un premier cas d'application.

La **troisième partie** met en œuvre l'approche proposée.

- Le **chapitre 6** décrit un outil informatique, développé sous la forme d'une application Web dynamique et mis en œuvre dans le cadre de cette thèse, afin de soutenir l'usage de la méthodologie SAMANTA par les ergonomes en charge de l'analyse et des spécifications d'interface de futurs systèmes. Le

chapitre 6 décrit également quelques spécificités techniques de cet outil.

- La méthodologie SAMANTA est appliquée au cas de la conception d'un système de drone dans le **chapitre 7**. A cette fin, nous avons utilisé l'outil décrit dans le **chapitre 6** pour la modélisation des tâches de l'opérateur et l'analyse de deux transitions spécifiques sélectionnées pour leur pertinence opérationnelle.

Ce chapitre présente également un autre cas d'application, issu d'une analyse réalisée par une ergonome de Bertin Technologies et apportant un regard extérieur sur l'outil et la méthodologie.

Enfin, le **chapitre 8** présente une conclusion synthétisant nos contributions. Nous y discutons des perspectives envisagées à la suite de nos travaux.



**PREMIÈRE PARTIE : TRANSITIONS & CONTEXTES : UNE REVUE DE
LA LITTÉRATURE**

2

Modéliser les transitions

Le dictionnaire Larousse définit le terme *transition* comme un « passage d'un état / une idée / une situation à un(e) autre ». A travers cette définition, le caractère changeant et évolutif de la transition est mis en avant, par opposition à un état statique. Dans notre contexte d'étude, les transitions entre tâches caractérisent cet aspect dynamique des systèmes étudiés ; qu'elles soient déclenchées par l'utilisateur ou par un tiers (être humain ou machine), et quelque soit l'impact de ces transitions sur le système et son utilisateur, la transition désigne à la fois l'enchaînement temporel entre les tâches du système considéré et les contextes associés.

Dans ce chapitre, nous abordons le sujet des transitions sous l'angle de *l'évolution temporelle* de l'activité de l'opérateur et nous intéressons aux travaux de recherche intégrant cette dynamique de changement de tâches utilisateur. Dans une première section, nous présentons différents modèles proposés dans le domaine de la modélisation des tâches et étudions la façon dont sont intégrées les transitions entre ces tâches (section 2.1). Dans un deuxième temps, nous aborderons deux approches où le concept de transition est omniprésent : l'approche cognitive des changements de tâches, étudiant les impacts cognitifs de ces transitions sur l'être humain (section 2.2) et l'étude d'un cas particulier de transition, les interruptions de tâches en Interaction Homme-Machine (section 2.3).

2.1 LES MODÈLES DE TÂCHES

L'analyse de l'activité de l'utilisateur constitue la première étape des processus de conception centrés utilisateur. Cette analyse permet d'appréhender toute l'étendue des tâches qui devront être intégrées au futur système. Des méthodologies telles que l'analyse de tâches basée sur les scénarios (Go et Carroll, 2003) ou l'analyse cognitive des tâches (*Cognitive Task Analysis*, CTA, Crandall *et al.* (2006)) permettent de systématiser cette étude de l'activité et la collecte de données associée, à travers des entretiens et l'observation de l'utilisateur en situation réelle ou expérimentale. Les modèles de tâches que nous présentons dans cette section sont autant de méthodes de représentations de l'activité de l'utilisateur, et les données représentées sont issues de méthodologies d'analyse de tâches (Diaper et Stanton, 2004). Nous proposons dans cette section d'étudier ces représentations et notamment la façon dont sont

représentées les transitions entre tâches.

2.1.1 Les différentes approches de modélisations de tâches

Les modèles de tâches proposés dans la littérature n'ont pas tous été construits dans le même but : certains sont voués à l'évaluation, d'autres à la définition des interactions entre l'homme et la machine. De plus, ils sont appliqués à la conception de systèmes allant des sites internet aux systèmes critiques. Certains auteurs ont proposé des comparaisons de ces modèles avec, à chaque fois, un objectif particulier, présentées dans la suite de cette section.

Ainsi, Limbourg et Vanderdonckt (2003) proposent une analyse comparative de 9 modèles, issus des domaines de la psychologie cognitive et ergonomie, de l'ingénierie logicielle, de l'allocation de tâches et de l'ethnographie. A partir de la méta-modélisation de chacun de ces modèles, ils les comparent sur des critères de différences syntaxique et sémantique. Leurs usages et points communs sont également mis en exergue. Chaque modèle est enfin comparé aux autres sur un plan « expressivité versus complexité » (figure 2.1), sur lequel nous observons que, pour les modèles considérés, l'expressivité est proportionnelle à la complexité du modèle : le choix d'une notation nécessite ainsi de faire un compromis entre ces deux paramètres.

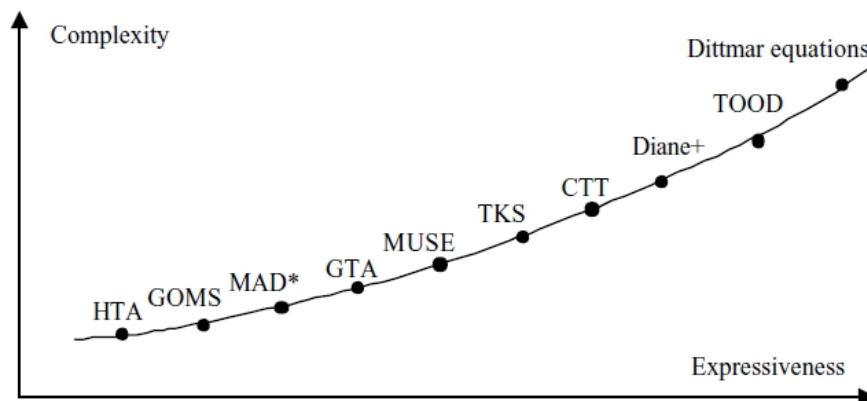


FIGURE 2.1 : expressivité versus complexité pour les modèles de tâches (Limbourg et Vanderdonckt, 2003).

Balbo *et al.* (2003) présentent une taxinomie en six axes, appliquée à la comparaison de six notations pour la modélisation d'un processus de réservation de vol sur Internet. Les catégories de la taxinomie proposée sont :

- Le but justifiant l'usage d'une notation, sélectionné parmi quatre propositions : « découverte & définition du système », « conception », « développement & déploiement » et « évaluation & prédiction » ;

- L'utilisabilité pour la communication : quel est la facilité de lecture et compréhension ? L'apprentissage par les néophytes est-il rapide ?
- L'utilisabilité pour la modélisation : est-elle facile à gérer, maintenir et manipuler par l'équipe de conception ?
- L'adaptabilité : si les objectifs ou le système cible changent, peut-on modifier les modèles facilement ?
- La « couverture » des différents concepts (pouvoir d'expression de la notation) : cet axe intègre la possibilité d'exprimer certains concepts à travers la notation, comme par exemple le caractère optionnel des tâches, la définition des types de tâches (système, utilisateur, interactive, abstraite), le degré de décomposition permis, la présence des relations temporelles, la possibilité d'intégrer des actions non standards ;
- L'extensibilité : la notation peut-elle intégrer de nouveaux concepts facilement ou est-elle fermée ?

Caffiau *et al.* (2010) se focalisent sur l'étude de six modèles de tâche à travers la comparaison des outils permettant de les soutenir. Cette approche pragmatique considère ainsi les modèles de tâches comme des moyens de communiquer et d'améliorer les échanges entre les utilisateurs et les concepteurs. L'ensemble des concepts proposés dans les différents outils sont passés en revue : le pouvoir d'expression des informations et caractéristiques des tâches, les aspects de planification locale (propres à une tâche donnée, comme les informations d'itération, de durée, etc.) et globale (les opérateurs temporels qui relient les tâches entre elles), la possibilité de définir des objets associés aux tâches et enfin, les différentes possibilités de validation automatique et simulation des outils.

2.1.2 L'intégration des transitions dans les modèles de tâches

Nous proposons dans cette section d'étudier une sélection de neuf modèles de tâches interactives, représentatifs du domaine de la conception centrée utilisateur et permettant de décrire les transitions entre ces tâches. Notre objectif n'est pas de fournir une description détaillée de ces modèles, mais d'en analyser les composantes relatives aux transitions entre tâches. Dans un premier temps, nous décrivons succinctement les modèles retenus pour notre étude, en précisant pour chacun la façon dont sont exprimées les transitions entre les tâches, à la fois sur l'aspect temporel et sur l'aspect logique (à travers les pré- et post-conditions). L'ensemble de ces données est ensuite présenté dans un tableau de synthèse.

2.1.2.1 HTA (*Hierarchical Task Analysis*)

La notation HTA (*Hierarchical Task Analysis*) (Annett et Duncan, 1967) est la plus ancienne notation de notre comparatif. Avant d'être une notation, HTA est une méthodologie d'analyse des tâches basée sur les entretiens et l'observation

des utilisateurs, ainsi que l'analyse de documents. Les résultats de cette analyse, initialement traduits sous forme textuelle, ont fait l'objet d'évolutions, permettant une représentation graphique des tâches (Shepherd, 1985).

La représentation des tâches dans la notation HTA est une décomposition strictement hiérarchique des tâches et sous-tâches. L'enchaînement temporel possible des différentes tâches est traduit par l'intermédiaire de « plans ». A chaque niveau hiérarchique de décomposition est associé un plan qui décrit, sous forme de texte et de manière informelle, les relations temporelles entre les tâches.

2.1.2.2 GOMS (*Goals, Operators, Methods, Selection rules*)

Issue du domaine de la psychologie cognitive, GOMS (*Goals, Operators, Methods, Selection rules*) (Card *et al.*, 1983) propose une modélisation du processeur humain afin d'évaluer de manière quantitative les performances d'un utilisateur. Afin d'atteindre les buts définis (*Goals*), l'utilisateur peut réaliser différentes opérations élémentaires physiques, perceptuelles ou cognitives (*Operators*), dont l'enchaînement est décrit dans les méthodes (*Methods*). Plusieurs méthodes peuvent mener au même but, le choix de la méthode se fait alors en fonction des règles de sélection (*Selection rules*). Plusieurs variantes de CMN-GOMS (le modèle initial) ont été proposées (John et Kieras, 1996) : GOMSL (*GOMS Language*), CPM-GOMS (*Cognitive-Perceptual-Motor GOMS*, CPM signifie également *Critical Path Method*) ou encore KML (*Keystroke Level Model*).

Ces modèles ne prennent pas en compte l'aspect temporel de l'enchaînement des tâches et s'appuient sur un enchaînement séquentiel des opérations élémentaires pour décrire l'atteinte d'un but (à travers les méthodes). Les règles de sélection permettent d'imposer des contraintes logiques sur l'exécution de certaines méthodes (à travers des relations SI...ALORS...SINON...).

2.1.2.3 GTA (*Groupware Task Analysis*)

Le modèle GTA (*Groupware Task Analysis*) (van der Veer *et al.*, 1996) permet de représenter l'activité des utilisateurs en intégrant l'aspect coopératif de l'environnement, introduisant le concept de rôle dans la description des tâches. Le modèle de tâches de cette notation se présente sous la forme d'une décomposition hiérarchique des tâches. Un opérateur unique décrit le lien entre les différentes sous-tâches d'une même tâche initiale.

L'opérateur de décomposition n'est pas formalisé dans la notation ; sa description au sein de l'outil Euterpe (soutenant cette notation) se fait par l'intermédiaire d'un champ entièrement textuel, laissant le concepteur choisir le type de décomposition qu'il souhaite attribuer (logique, temporelle...). Le formalisme permet également l'usage de pré- et post-conditions pour l'exécution des tâches.

2.1.2.4 UAN (*User Action Notation*)

La notation UAN (*User Action Notation*) (Hartson et Gray, 1992) formalise, sous forme de texte, la description des tâches élémentaires de manière non hiérarchique. Chaque tâche est décrite dans un tableau à trois colonnes correspondant aux trois aspects de la tâche : les actions utilisateurs (*user actions*), les retours de l'interface (*interface feedbacks*) et le nouvel état de l'interface (*interface state*). Les tâches décrites sont élémentaires (sélection, suppression de fichier...). Plusieurs extensions de cette notation ont été proposées, comme par exemple NUAN (*New User Action Notation*) (Venema, 1999), qui est intégrée à GTA pour la description de ses tâches élémentaires, et XUAN (*eXtended User Action Notation*) (Gray *et al.*, 1994), qui intègre des pré- et post-conditions et la possibilité de décrire les aspects temporels des tâches de manière plus explicite.

2.1.2.5 TKS (*Task Knowledge Structure*)

A travers la notation TKS (*Task Knowledge Structure*) issue du domaine de la psychologie cognitive (Johnson *et al.*, 1988), l'analyste va manipuler une structure modélisant toutes les connaissances d'un utilisateur au sujet d'une tâche : les rôles, buts, plans et procédures (constituées d'objets et d'actions). A chaque tâche est associé un but et inversement. Une structure de buts et sous-buts (et donc des tâches et sous-tâches) peut être décrite à travers une décomposition hiérarchique des tâches, ou par des « constructeurs » temporels ou causaux.

2.1.2.6 Diane+

Diane+ (Tarby et Barthet, 1996) est une méthodologie ayant pour but de rapprocher les domaines des facteurs humains et de l'ingénierie logicielle. Le modèle de tâches intégré décrit les relations temporelles et hiérarchiques entre les tâches utilisateurs. La décomposition des sous-tâches est visualisée dans des boîtes : à chaque tâche correspond une boîte de sous-tâches, liées entre elles par un opérateur unique (temporel ou logique).

- Lorsque les sous-tâches sont exécutées en parallèle, aucun opérateur n'apparaît dans la boîte.
- Lorsqu'elles sont exécutées en séquence, les sous-tâches sont reliées entre elles par des flèches, indiquant l'ordre de réalisation.
- D'autres opérateurs peuvent spécifier la combinaison logique des sous-tâches au sein de la même boîte : XOR (une des sous-tâches est exécutée), OR (au moins une des sous-tâches est exécutée), AND (toutes les sous-tâches sont exécutées).

Ainsi, chaque bloc de sous-tâches ne pouvant être associé qu'à un seul opérateur, il peut être nécessaire de créer des tâches factices pour permettre de combiner plusieurs

opérateurs. L'outil TAMOT associé à cette notation permet également de spécifier des pré-conditions sous forme de texte.

2.1.2.7 CTT (*Concurrent Task Tree*)

La notation CTT (*Concurrent Task Tree*) (Paternò *et al.*, 1997) permet de décrire la décomposition hiérarchique et l'enchaînement temporel des tâches d'un système interactif. L'aspect coopératif du système est pris en compte à travers la définition de rôles utilisateurs, auxquels sont associés autant d'arbres de tâches. Les relations temporelles entre les tâches d'un même niveau d'abstraction sont décrites de manière formelle à partir des opérateurs LOTOS (ISO, 1989) : aux quatre opérateurs « classiques » (séquence, concurrence, choix et ordre indépendant) s'ajoutent l'interruption et la désactivation (arrêt d'une tâche en cours d'exécution) ainsi que deux variantes de la séquence et du choix intégrant la transmission d'information. Le modèle permet également la définition de pré-conditions (présentées sous forme de texte dans l'éditeur associé, CTTE).

Certains modèles plus récents s'appuient sur la notation CTT, nous notons en particulier CTML (*Collaborative Task Modelling Language*) (Sinnig *et al.*, 2007), associant UML (*Unified Modelling Language*) et CTT, ainsi que COMM (Collaborative et MultiModale) (Jourde *et al.*, 2010), une extension directe de CTT. Ces deux formalismes enrichissent la notation CTT en y adjoignant de nouveaux opérateurs.

CTML étend la signification de l'opérateur d'alternative (\square) en précisant si le choix est déterministe (choix issu de l'utilisateur, représenté par le symbole $\square D$) ou non-déterministe (choix décidé par le système, représenté par le symbole $\square N$). De même, CTML introduit l'opérateur d'itération concurrente (symbolisée $\#$ au lieu de $*$) afin de marquer une tâche comme pouvant être réalisée plusieurs fois en parallèle (par différents utilisateurs par exemple).

COMM reprend le concept d'itération concurrente de CTML et définit un ensemble d'opérateurs unaires basés sur les deux opérateurs initiaux (itérations séquentielle $*$ et concurrente $\#$) afin de préciser la cardinalité des instances de la tâche sous forme de chiffres. Afin de préciser les différentes relations temporelles entre les tâches, COMM donne la possibilité à l'analyste d'utiliser des opérateurs issus des relations de Allen. Jourde (2011) note néanmoins que ce niveau de précision « semble peu pertinent » entre les tâches d'un niveau d'abstraction élevé.

2.1.2.8 MAD (Méthode Analytique de Description des tâches)

La notation MAD (Méthode Analytique de Description des tâches) (Scapin et Pierret-Golbreich, 1989) décompose les tâches utilisateurs de manière hiérarchique. La relation temporelle entre les tâches est également intégrée au niveau des tâches parentes (de la même manière que dans GTA par exemple). L'extension MAD*

(MAD-STAR : Modèle Analytique de Description de tâches orienté spécification d'interfaces) (Bastien et Scapin, 1993) ajoute notamment la description sémantique des objets.

A travers le modèle **K-MAD** (*Kernel of Model for Activity Description*), Lucquiaud (2005a) propose de définir un modèle de description des tâches orienté utilisateurs, intégrant les données jugées comme essentielles à la description de l'activité issues de l'étude de plusieurs autres modèles. Ce « noyau » intègre la description des tâches, des utilisateurs, des événements et des objets. La description graphique et le formalisme retenu pour l'enchaînement temporel sont les mêmes que ceux utilisés dans MAD*. Les relations temporelles sont spécifiées au niveau de la tâche parente et sont partagées par toutes les sous-tâches. K-MAD est soutenu par un éditeur, K-MADe.

2.1.2.9 AMBOSS

AMBOSS (Giese *et al.*, 2008) est avant tout un environnement de modélisation des tâches, dont nous n'avons pu trouver la description détaillée. Nous l'intégrons néanmoins à notre comparatif car il est proche d'autres notations du domaine et est référencé à plusieurs reprises dans des publications récentes d'auteurs différents.

AMBOSS a été conçu pour prendre en compte la sûreté des systèmes critiques complexes. La décomposition hiérarchique des tâches suit le modèle de K-MAD, où les tâches et sous-tâches sont représentées sous forme d'arbre, et les relations temporelles sont spécifiées au niveau des nœuds, s'appliquant à toutes ses sous-tâches. Les pré-conditions sont présentes sous la forme de « barrières ».

2.1.2.10 TOOD

La méthode TOOD (Task Object Oriented Design) Abed *et al.* (2003) est un processus de conception de systèmes complexes reposant sur deux modèles : un modèle structurel statique et un modèle structurel dynamique. Le modèle structurel statique décrit la décomposition hiérarchique des tâches de l'utilisateur en s'appuyant sur le formalisme SADT-Extended (Feller et Rucker, 1989). Chaque tâche est décrite comme un objet pour lequel sont spécifiés les interfaces d'entrée et de sortie, les ressources associées à la tâche ainsi que le corps de la tâche (les sous-tâches ou la procédure mise en œuvre, selon le type de tâche).

Le modèle structurel dynamique complète la version statique en décrivant l'enchaînement logique et temporel des tâches à travers une structure de contrôle basée sur les réseaux de Petri « orientés objets » (*Object Petri Net* ou OPN) (Sibertin-Blanc, 1985), permettant d'exprimer les enchaînements à la fois graphiquement mais aussi sous la forme d'expressions mathématiques (textuelles).

2.1.3 Comparaison des notations

Le tableau 2.1 présente une vue d'ensemble des modèles passés en revue. Ce tableau se présente en quatre parties, pour chaque modèle nous précisons :

- les informations générales telles que le type de représentation utilisée (textuelle ou graphique), l'existence d'un éditeur disponible en ligne et la référence bibliographique associée à chaque notation ;
- les informations de planification locale (reprenant la définition de Caffiau *et al.* (2010)) c'est-à-dire les attributs qui permettent de caractériser la tâche indépendamment de ses sous-tâches (durée, caractère optionnel, interruptible et itératif) ;
- les informations de planification globale, décrivant les enchaînements temporels entre les tâches. Nous caractérisons la possibilité de définir des relations de base entre tâches, à savoir la « séquence », le « choix », l'« ordre indépendant », et la « concurrence » (parfois appelée « parallélisme »), ainsi que la possibilité de définir des relations avancées à travers des opérateurs spécifiques ;
- la possibilité de définir des pré-conditions (conditions de début de tâche) et post-conditions (conditions de fin de tâche) pour l'exécution des tâches. Ces conditions sont généralement associées à des objets définis dans la notation, qui peuvent décrire des concepts abstraits ou concrets.

L'ensemble des modèles de tâches comparés dans le tableau 2.1 permet de décrire les transitions entre les tâches, avec des niveaux de précisions et de formalisation plus ou moins élevés.

Concernant les notations « textuelles », les relations temporelles (rubrique « planification globale ») ne sont généralement pas formalisées, excepté pour UAN, qui définit un ensemble d'opérateurs permettant de formaliser ces relations, et TOOD qui permet d'exprimer ces relations sous formes mathématiques. Les autres notations expriment ces transitions à travers les plans (ou équivalent) décrivant l'enchaînement escompté des tâches ;

Pour les notations « graphiques », les opérateurs de base sont toujours présents. Nous notons que toutes les notations sauf CTT expriment les relations temporelles entre les sous-tâches au niveau de la tâche parente, tandis que CTT permet de visualiser individuellement chaque relation temporelle entre deux tâches de même niveau hiérarchique.

- Dans CTT, des ambiguïtés sur l'ordre d'enchaînement des opérateurs temporels peuvent apparaître. Pour y remédier, Paternò *et al.* (1997) proposent soit de définir un ordre de priorité entre les opérateurs temporels soit de créer des tâches intermédiaires permettant de lever les ambiguïtés, en ajoutant des niveaux d'abstraction.
- A l'inverse, lorsque la logique temporelle est décrite au niveau de la tâche parente, plusieurs niveaux de décomposition peuvent être nécessaires, imposant à l'analyste la création de ces tâches intermédiaires ne portant pas de signification réelle dans l'activité de l'utilisateur : elles ne servent qu'à « factoriser »

CHAPITRE 2. MODÉLISER LES TRANSITIONS

TABLEAU 2.1 : comparaison de neuf notations de modélisation des tâches sur leur caractéristiques temporelles et causales.

Notation	HTA	GOMS	GTA	UAN	TKS	Diane+	CTT	K-MAD	AMBOSS	TOOD
Représentation	Textuelle & graphique	Textuelle	Graphique	Textuelle	Textuelle	Graphique	Graphique	Graphique	Graphique	Textuelle & Graphique
Éditeur associé	-	-	Euterpe	-	-	TAMOT	CTTE	K-MADe	AMBOSS	(TOOD-IDE)
Référence	Annett et Duncan (1967)	Card et al. (1983)	van der Veer et al. (1996)	Hartson et Gray (1992)	Johnson et Johnson (1988)	Tarby et Barthet (1996)	Paterno et al. (1997)	Lucquiaud (2005)	Giese et al. (2008)	Abed et al. (2003)
Planification locale										
Durée	Sous forme de texte (dans les plans)	✓	-	✓	Non formalisés, mais peut être une connaissance sur la tâche	-	✓	✓	✓	✓
Optionnelle		-	-	✓		✓	-			
Interruptible		-	-	Opérateur →		-	Opérateur >	Booléen	-	Fonction de priorité
Itérative		-	-	✓		✓	✓	✓	-	✓
Planification globale										
Relations de base	Idem	Séquence et concurrence	Non formalisés, libellé au choix	✓	Constructeurs temporels non formalisés	✓	✓	✓	✓	✓
Relations avancées		-	Interruption Attente	-		Interruption, désactivation	-	Réseaux de Petri		
Enchaînements logiques										
Pré-condition	Idem	✓	✓	Ajoutées dans XUAN (Gray et al., 1994)	Constructeurs logiques non formalisés	✓	✓	✓	✓	✓
Post-condition		✓	✓			-	✓	-	✓	

l'arbre.

Seules CTT et UAN intègrent des relations temporelles avancées : tous deux intègrent l'interruption (avec reprise possible de la tâche interrompue). CTT propose également une variante de l'interruption, appelée « désactivation », qui consiste à une interruption sans reprise (et s'apparente donc à une « séquence » où la tâche initiale n'a pu se terminer correctement). UAN intègre une relation permettant de définir un temps d'attente entre deux tâches (où rien ne se passe). TOOD permet d'utiliser la puissance (et la complexité) des réseaux de Pétri afin d'exprimer également ce type relations avancées. Nous n'avons pas intégré dans ce tableau les notations CTML et COMM (§ 2.1.2.7) car issues de CTT, mais nous notons qu'elles proposent des relations temporelles avancées supplémentaires.

Enfin, nous notons que parmi les notations présentées, celles qui s'expriment sous une forme graphique sont soutenues par des outils disponibles en ligne, à l'exception de TOOD-IDE, non maintenu. Le côté graphique de ces notations en facilite la création et les échanges avec les utilisateurs autour des modèles de tâches, l'existence d'outils pour appuyer cette phase de création assure la diffusion de la notation.

2.1.4 Commentaire sur les modèles de tâches

En synthèse de cette section, nous notons que la majorité des modèles de tâches intègre dans leur notation la description des transitions entre tâches, de façon plus ou moins formelle et à des niveaux de détails variés. Néanmoins, la description des transitions entre tâches ne constitue pas le cœur de ces modèles, dont les buts initiaux n'intègrent pas l'étude de ces transitions. Celles-ci apparaissent plutôt comme des concepts secondaires constituant, au mieux, des objets liant les tâches entre elles comme dans le cas de CTT. Le plus souvent, la caractérisation de l'ordonnancement temporel entre les tâches se présente comme un attribut de la tâche, servant à sa caractérisation et à celle de ses sous-tâches, au même rang que d'autres informations telles que le nom, la fréquence, le but ou plate-formes d'exécution.

2.2 LE TASK SWITCHING, OU L'APPROCHE PAR LE « COÛT COGNITIF »

L'impact cognitif des changements de tâches sur l'être humain a été largement étudié dans le domaine de la psychologie cognitive depuis le milieu des années 1990. Ce domaine de recherche, classé sous les termes anglais de *task switching* ou *task shifting*, s'attache à l'étude des transitions entre tâches dites « simples » (§ 2.2.1.1). Les expérimentations réalisées dans ce contexte, souvent menées en laboratoire, placent le sujet dans un cadre expérimental assez éloigné du contexte opérationnel que nous abordons dans le reste de notre étude. Néanmoins, nous proposons dans cette section un aperçu des différentes théories relatives à ce domaine, qui présentent des résultats relatifs mais intéressants, et dont nous pourrions nous inspirer dans nos travaux futurs. Nous nous appuyerons notamment sur les articles de Logan (2003);

Monsell (2003); Kiesel *et al.* (2010), qui font état des différentes tendances dans ce domaine.

2.2.1 Préambule

En préambule, nous définissons les concepts nécessaires à la compréhension des théories du domaine du *Task Switching*.

2.2.1.1 Les tâches

Les tâches mises en jeu dans les expérimentations liées aux études sur le changement de tâches ont un niveau d'abstraction que l'on qualifie de « bas ». Celui-ci correspond à l'échelon le plus bas des niveaux d'abstraction de Cooper et Shallice (2000). Ces tâches se décomposent en deux étapes : l'affichage d'un stimulus (correspondant à la tâche à réaliser) et la réponse du sujet (qui réalise une opération mentale et l'action motrice associée). Ces tâches consistent généralement en des exercices d'addition ou de soustraction (Jersild, 1927), de la détection de couleurs ou de types de lettre (voyelle/consonne) (Rogers et Monsell, 1995; Allport *et al.*, 1994, entre autres) ou bien de tests de Stroop (1935), illustré sur la figure 2.2.



FIGURE 2.2 : exemple de tâche de Stroop (1935), consistant soit à lire le mot soit à nommer la couleur correspondant au mot. Pour chaque stimulus – l'affichage du mot « vert » écrit en orange par exemple – le sujet devra activer la réponse correspondant à la tâche (lire le mot / énoncer sa couleur).

2.2.1.2 Le concept de *task-set*

Le concept de *task-set* (Jersild, 1927; Allport *et al.*, 1994; Rogers et Monsell, 1995) correspond à la préparation cognitive du sujet à la tâche à venir. De la même façon qu'un cycliste changera de vitesse avant de gravir une colline à vélo, un certain nombre de pré-requis mentaux doivent être activés afin que le sujet puisse remplir avec succès cette nouvelle tâche. Ce *task-set* inclut « la représentation de l'objectif de la tâche » (par exemple *définir la couleur du stimulus*), un « ensemble de stimuli pertinents pour la tâche » (par exemple *bleu* et *rouge*), un « ensemble de réponses possibles » (par exemple, *appuyer sur la flèche droite ou la flèche gauche du clavier*) et une « association entre les stimuli et les réponses (*mapping*) » (Koch *et al.*, 2010). La difficulté de la tâche repose généralement dans le *mapping* entre les différents

concepts : dans l'exemple de la figure 2.2, il est plus facile de lire le mot lui-même que le nom de la couleur dans lequel il est écrit.

2.2.1.3 Le coût cognitif de la transition ou *switch cost*

Dans la littérature, l'impact cognitif des transitions entre tâches est mesuré à travers le *switch cost*, métrique intégrant deux mesures expérimentales distinctes :

- le **temps de réponse** (*response time* ou RT), temps entre le stimulus et la réponse du sujet. Ce temps varie typiquement de quelques dizaines à quelques centaines de millisecondes selon la tâche et la configuration de l'expérimentation.
- la **validité des réponses** du participant, présentée sous la forme d'un taux d'erreur.

Dans ces expérimentations, la comparaison porte généralement sur la différence (ou non) des valeurs de *switch cost* lors qu'une même tâche est répétée ou lorsque deux tâches alternent (entraînant un changement de tâches).

2.2.1.4 Les paradigmes expérimentaux

Jersild (1927) est considéré comme le premier à avoir mené des expérimentations afin de mesurer l'impact des changements de tâches sur le temps de réponse. Le paradigme expérimental qu'il propose consiste à mesurer le temps de réalisation de deux tâches A et B selon deux configurations : répétition en bloc d'une même tâche (AAAA ou BBBB) ou alternance entre les deux tâches (ABAB) au sein d'un même bloc (condition de *task switching*). Les tâches consistent à ajouter ou soustraire un nombre prédéfini à une série de nombres. Les participants (des étudiants de niveau universitaire), effectuent ces tâches en notant les résultats sur une feuille de papier, allant même parfois jusqu'à se chronométrer eux-mêmes. Les conclusions de Jersild (1927) indiquent que le temps de réponse des participants augmente en situation d'alternance, et lorsque la complexité de la tâche augmente (ajouter 17 à un nombre à deux chiffres est considéré plus difficile qu'ajouter 6). Allport *et al.* (1994) proposent une variante du paradigme expérimental de Jersild, basé sur différentes variations des stimuli de Stroop (Stroop, 1935; MacLeod, 1991).

Rogers et Monsell (1995) critiquent le montage de Jersild car les deux séries (répétition ou alternance) sont indépendantes, ce qui expliquerait que la charge cognitive soit plus importante dans le cas de l'alternance (car deux *task-sets* sont présents en mémoire). Ils utilisent en conséquence le paradigme de « séries alternées » (*alternating-runs paradigm*) qui consiste à répéter la tâche A n fois, suivie de n répétitions de la tâche B et ainsi de suite (pour $n = 2$, on aurait alors un motif global de la forme AABBAABB...). Ce paradigme expérimental permet d'étudier à la fois les répétitions et les alternances au sein d'une même série.

Un troisième paradigme présent dans la littérature consiste à indiquer avant chaque essai la tâche à venir, à l'aide d'une indication (souvent visuelle). Les tâches s'enchaînent ainsi de manière aléatoire, mais les différents *task-set* doivent être présents en mémoire (Meiran, 1996).

Enfin, deux autres paradigmes sont également cités dans la littérature. Le premier est le paradigme des « instructions intermittentes » (Allport et Wylie, 2000; Altmann et Gray, 2008) reprenant l'idée des séries alternées, mais l'ordre prédéfini des blocs est remplacé par la présentation d'un indice tous les n essais, indiquant la tâche à réaliser pour les n essais suivants. Le dernier paradigme consiste à laisser le sujet choisir la tâche qu'il va réaliser, le choix se fait au moment de la présentation des stimuli, qui sont nécessairement bivalents : le même stimulus permet de réaliser deux tâches, comme les stimuli de Stroop (Arrington et Logan, 2004).

2.2.2 Les interprétations du « coût de changement »

Les paradigmes expérimentaux présentés en introduction ont amené les chercheurs à différentes interprétations des phénomènes observés. Deux courants co-existent depuis le début des années 1990, chacun proposant une interprétation de la présence d'un coût cognitif lors des transitions. La première explique le *switch cost* par les effets d'interférence des tâches précédentes sur les nouvelles tâches : il s'agit de la théorie de « l'inertie du *task-set* » (*task-set inertia*), introduite par Allport *et al.* (1994). La deuxième école explique l'existence du *switch cost* par la nécessité du sujet à se préparer à la tâche à venir lors d'un changement de tâches : il s'agit de la théorie de la « reconfiguration du *task-set* » (*task-set reconfiguration*) de Rogers et Monsell (1995).

Les études récentes s'accordent sur le fait que ces deux sujets de recherche – la préparation à la tâche à venir et l'interférence de la tâche passée – permettent d'expliquer le coût cognitif lié au changement de tâches et ne sont pas incompatibles l'un par rapport à l'autre. Nous présentons dans la suite de cette section les expérimentations à l'origine des courants de recherche associés, ainsi que les effets qui ont pu être mis en exergue lors de ces expérimentations et pouvant être pertinents dans le cadre de notre sujet de recherche.

2.2.2.1 Les effets d'interférences

Task-Set Inertia (TSI)

Les premiers travaux menés sur les effets d'interférence dans le changement de tâches ont introduit le concept de l'inertie du *task-set*. Ces travaux, menés par Allport *et al.* (1994, expérimentations 4, 5 et 7), expliquent que lorsque le sujet a déjà réalisé la tâche au cours de la même série (de type ABA), l'impact du retour à la première tâche est négatif (*switch cost* plus élevé que pour une série AAA), traduisant

un effet d'interférence du *task-set* lié à « l'inertie » de ce dernier à travers les essais. En d'autres termes, dans la séquence ABA, la deuxième occurrence de la tâche A est influencée par le fait que le sujet a déjà réalisé cette tâche une première fois. Cet effet se dissipe après quelques minutes sans exécution de la tâche concernée.

Asymétrie du coût de transition et effet d'inhibition

Ce type d'interférence est appuyé par les expérimentations montrant que le coût cognitif de transition est asymétrique (Allport et Wylie, 2000; Allport *et al.*, 1994) : les transitions d'une tâche simple (lire un mot, par exemple) vers une tâche plus compliquée (nommer la couleur dans lequel est écrit ce mot) induisent un coût de changement moins important sur le sujet que la transition inverse. Ce phénomène serait expliqué par la théorie de l'inertie du *task-set* : les tâches plus difficiles correspondent à une préparation cognitive plus « forte », qui persisterait plus longtemps en mémoire, venant troubler le déroulement des tâches plus faciles (Allport *et al.*, 1994).

La théorie de l'inhibition du *task-set* (Mayr, 2009; Mayr et Keele, 2000; Koch *et al.*, 2010) montre, en introduisant une troisième tâche dans les expérimentations (Mayr et Keele, 2000), qu'il est plus coûteux (en terme de *switch cost*) de revenir à une tâche (série ABA) que de l'oublier (série CBA), traduisant un effet d'inhibition de la première tâche lors du passage à la deuxième, et la nécessité d'une réactivation de la tâche lorsqu'elle doit être à nouveau exécutée (en troisième position). Kiesel *et al.* (2010) concluent que les processus d'inhibition de même que l'effet de l'inertie de l'activation d'un *task-set* sont tous les deux des causes d'interférence lors de transitions de tâches.

Interférence des stimuli

Certaines expérimentations ont également permis de montrer l'influence des stimuli sur le *switch cost*. Le stimulus est considéré comme composante exogène dans la préparation cognitive, son apparition déclenche la préparation du *task-set*. Nous relevons trois effets d'interférence générés par les stimuli :

- Dès les premières études sur le sujet (Jersild, 1927; Allport *et al.*, 1994; Rogers et Monsell, 1995), les expérimentations ont permis de conclure que les *stimuli univalents induisent un coût de transition plus faible*. L'univalence d'un stimulus signifie que chaque stimulus est associé à une tâche différente (à l'inverse du test de Stroop, où le même stimulus peut induire deux tâches, on parle alors de stimulus bivalent). Par exemple, l'alternance entre la lecture d'un mot et l'addition de nombres constitue une alternance de deux stimuli (mots et nombres) univalents.
Ainsi, relevons que la présentation de deux stimuli différents permet de réduire le coût de transition (notons tout de même que les tâches exécutées dans ces expérimentations sont connues à l'avance par les sujets).
- L'effet de *congruence* des stimuli correspond au fait que les sujets répondent plus rapidement lorsque le stimulus (nécessairement bivalent) présenté est as-

socié à la même réponse pour les deux tâches en jeu. Par exemple, si on associe une touche du clavier à la classification d'un nombre comme pair (tâche A) ou supérieur à 50 (tâche B), et une autre touche aux autres conditions (nombre impair, nombre inférieur à 50), le temps de réponse sera meilleur sur des essais où est présenté un nombre pair supérieur à 50 ou impair inférieur à 50 (Rogers et Monsell, 1995; Wendt et Kiesel, 2008; Kiesel *et al.*, 2010).

- L'effet de *crosstalk* est issu de la comparaison du *switch cost* entre des essais où les deux stimuli associés aux deux tâches sont affichés en même temps et des essais où le deuxième stimulus est remplacé par un stimulus neutre. Les résultats de Rogers et Monsell (1995, expérimentation 1) montrent que lorsque les deux stimuli sont présentés, le sujet traite également le stimulus non pertinent, induisant ainsi un coût de transition plus élevé pour cette condition.

Interférence des réponses

D'autres études portent notamment sur l'influence des réponses et de leur association aux stimuli (l'objet des tâches). Ces études mesurent la différence de *switch cost* lorsque la réponse à deux stimuli se fait à travers les mêmes touches (bivalence de la réponse). Par exemple, la flèche gauche permet à la fois de classer un chiffre comme pair (tâche A) et une lettre comme une consonne (tâche B), tandis que la flèche de droite permet de classer les chiffres comme impairs pour la tâche A et les lettres comme des voyelles pour la tâche B. Comme pour le cas de la bivalence des stimuli, la bivalence des réponses induit un coût de transition plus élevé : chaque stimulus doit être associé à une réponse différente, afin de limiter le coût de la transition (Brass *et al.*, 2003; Meiran, 2000).

2.2.2.2 Les effets de préparation

Task-Set Reconfiguration (TSR)

Les travaux menés sur les effets de préparation ont été introduits par Rogers et Monsell (1995). Ils justifient l'augmentation de la mesure de *switch cost* lors d'un changement de tâches par l'effet de reconfiguration mentale nécessaire au sujet pour se préparer à la tâche à venir. Ces résultats sont soutenus par des expérimentations au cours desquelles le *switch cost* est mesuré pour différentes valeurs de l'intervalle réponse-stimulus (le temps entre la réponse à un stimulus et l'apparition du stimulus suivant). Rogers et Monsell (1995, expérimentations 2, 3 et 4) montrent que le *switch cost* est plus important lorsque cet intervalle est court, et en augmentant la durée de l'intervalle réponse-stimulus, le *switch cost* diminue.

Néanmoins, la valeur du temps de réponse atteint une asymptote et ne décroît plus lorsque la durée de l'intervalle réponse-stimulus est supérieure à une seconde (le taux d'erreur est alors proche de zéro), introduisant un « coût cognitif résiduel » (*residual cost*). Monsell *et al.* (2000) expliquent ce coût résiduel par l'exécution d'un processus de contrôle exogène (déclenché par l'apparition du stimulus) nécessaire pour terminer la reconfiguration mentale du *task-set* (TSR). Ainsi, la théorie de la

reconfiguration (TSR) se décomposerait en deux processus exécutifs distincts lors d'un changement de tâches (figure 2.3) :

- un premier processus de contrôle endogène (déclenché par le sujet en prévision du changement) dont l'influence sur le coût de changement diminue lorsque l'intervalle de préparation augmente ;
- un deuxième processus de contrôle exogène, qui expliquerait le coût résiduel car déclenché uniquement après l'apparition du stimulus.

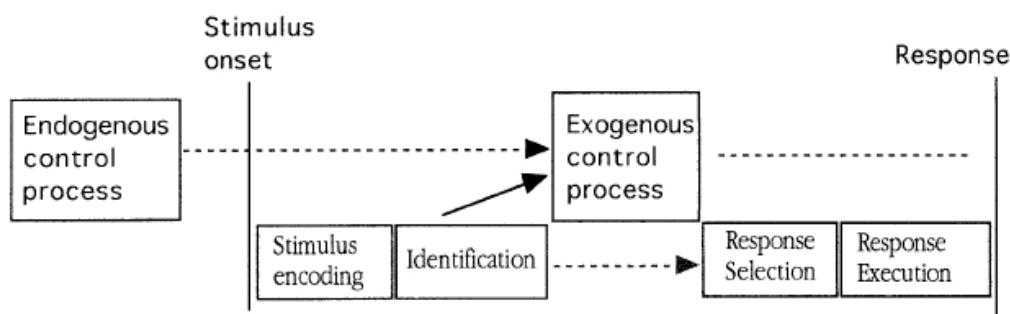


FIGURE 2.3 : la décomposition des processus exécutifs lors d'un changement de tâches (Monsell *et al.*, 2000). Les processus de contrôle endogène (pré-stimulus) et exogène (post-stimulus) ne sont pas exécutés lorsqu'il n'y a pas de changement de tâches.

Effets des indices sur la préparation

Quelques résultats issus de la littérature portent sur l'effet des indications sur la préparation :

- Rogers et Monsell (1995, expérimentation 5) ont montré que l'ajout d'un signal « neutre » (un cadre vide à l'endroit d'apparition du prochain stimulus) avant de présenter le stimulus n'a qu'un effet très faible sur le coût de la transition.
- Afin de dissocier l'impact sur le *switch cost* du changement de tâches et celui du changement d'indication avant la tâche (puisque une tâche différente implique une indication différente), Mayr et Kliegl (2003); Logan et Bundesen (2003); Arrington *et al.* (2007) proposent un paradigme expérimental où deux indices correspondent à une même tâche. Les résultats de ces études suggèrent que quelque soit le type de transition (répétition ou alternance), le temps de réponse augmente lorsque deux indices différents sont présentés pour une même tâche.

Ainsi, il est préférable de n'associer qu'un indice à une tâche donnée.

Caractère aléatoire de la reconfiguration

Enfin, De Jong (2000) notamment suggère que la reconfiguration du *task-set* est un processus aléatoire : selon les essais, le sujet va s'engager dans un tel processus ou non. Cette théorie est parfois reprise dans d'autres études pour justifier le coût

résiduel non expliqué, notamment dans la théorie de l'activation de Altmann et Gray (2008), qui intègrent ce caractère aléatoire à leur modélisation des processus cognitifs en jeu.

2.2.3 Commentaire sur le Task Switching

Quelle que soit l'explication sous-jacente, l'effet au niveau cognitif des changements de tâche est indéniable et a été prouvé à maintes reprises. Les travaux de recherche actuels dans ce domaine reconnaissent les différentes théories présentées dans cette section, et les différents auteurs ne les considèrent plus comme incompatibles. La limite de ces travaux est l'aspect expérimental des tâches présentées : les résultats obtenus sont valides sur des tâches « de laboratoire », qui sont éloignées de l'aspect opérationnel des systèmes abordés dans ce mémoire.

2

2.3 LES INTERRUPTIONS EN INTERACTION HOMME-MACHINE

Nous avons défini la transition comme le passage d'une tâche A à une tâche B. L'interruption d'une tâche A par une tâche B peut être vue comme un cas particulier de transition : il s'agit de la composition d'une transition de la tâche A vers la tâche B et de la transition inverse, de la tâche B vers la tâche A. Ainsi, lors d'une interruption, et à la différence peut-être d'une transition plus « classique », la tâche initiale A sera également la tâche finale. Il est donc nécessaire pour l'être humain d'adopter des comportements particuliers pour faciliter l'ensemble du processus de changements de tâches.

Cette section offre un aperçu de la littérature sur les interruptions. Dans un premier temps, nous revenons sur le concept d'interruption à travers une définition plus globale, proposée par McFarlane & Latorella. Puis nous présentons le sujet du point de vue de la psychologie cognitive, en particulier l'effet de préparation avant l'interruption et de reprise de la tâche conservée en mémoire. Enfin, nous présentons quelques travaux du domaine de l'interaction homme-machine portant sur la coordination entre le système interruptif et l'être humain ainsi que l'intégration des interruptions en phase de conception.

2.3.1 Taxinomie de l'interruption humaine

Dans un rapport de synthèse sur le thème des interruptions dans le domaine de l'Interaction Homme-Machine, McFarlane (1997) propose la définition suivante de l'interruption, issue de l'étude de treize définitions du terme « interruption » dans autant de domaines :

“

L'interruption humaine est le processus de coordonner les changements brusques dans l'activité des personnes

(McFarlane, 1997)

”

Nous remarquons que dans la définition donnée par McFarlane, l'interruption ne concerne pas forcément celle d'une tâche A par une tâche B comme nous l'avons introduit. Cette définition est complétée par une taxinomie, décrivant les différents aspects des interruptions.

2

2.3.1.1 Présentation de la taxinomie de McFarlane

La taxinomie proposée par McFarlane (1997) identifie huit sujets d'étude liés aux interruptions. L'auteur présente cette taxinomie comme un outil théorique d'investigation des interruptions à travers ces huit axes de recherche. Nous décrivons dans la suite ces huit catégories (McFarlane, 1997; McFarlane et Latorella, 2002).

Source de l'interruption (*Source of Interruption*)

La source de l'interruption correspond à l'entité qui déclenche cette interruption. Il peut s'agir d'une source interne (aspect endogène de l'interruption), l'humain décide alors d'interrompre sa tâche courante sans intervention d'un tiers, ou externe (aspect exogène). La source externe d'interruption peut être un être vivant (un autre être humain, un animal) ou non (la fenêtre modale d'une application de notre ordinateur, la sonnerie du réveil, une avalanche de neige interrompant une descente à ski...).

Caractéristiques individuelles de la personne interrompue

(*Individual Characteristic of Person Receiving Interruption*)

Cette catégorie concerne l'étude des caractéristiques cognitives et physiques de l'individu qui est interrompu : motivation dans l'interruption (pour effectuer la nouvelle tâche ou reprendre la tâche interrompue), facilité à exécuter plusieurs tâches en même temps (capacité à être « multitâche »), habitude à exécuter les tâches (que ce soit la tâche interrompue ou celle qui interrompt), volonté d'être interrompu (peut par exemple dépendre de la source de l'interruption, l'interruption par un supérieur hiérarchique sera peut-être mieux acceptée que par un étranger).

Méthode de coordination (*Method of Coordination*)

Cette catégorie s'intéresse aux différentes méthodes permettant de à la personne interrompue de coordonner l'interruption de la tâche principale. Outre le fait qu'il puisse ne pas y avoir de coordination (interruption immédiate), les méthodes proposées sont la négociation en temps réel (lors d'un dialogue avec

un collègue de travail par exemple), l'interruption à travers un médiateur (qui décide du meilleur moment pour interrompre), les interruptions programmées (par un planning prévu à l'avance, à certains intervalles de temps par exemple). Ces méthodes de coordination font l'objet de plusieurs études. La section 2.3.3 aborde ce sujet, appliqué au domaine de l'interaction homme-machine.

Sens de l'interruption (*Meaning of Interruption*)

Cette catégorie rassemble les études portant sur les raisons des interruptions, le « pourquoi ». La taxinomie de McFarlane (1997, tableau 13) répertorie de nombreuses significations pour les interruptions : elles peuvent être le produit d'une alerte à l'utilisateur, d'une simple notification, d'un rappel sur l'occurrence d'un événement, d'une suggestion, d'une transmission ou une demande d'informations, d'une tâche supplémentaire à réaliser, etc. Parfois, l'interruption ne porte pas de sens supplémentaire, outre le fait qu'elle interrompt la tâche, comme lors d'une panne.

Méthode d'expression (*Method of Expression*)

Afin de réduire l'impact négatif d'une interruption, la méthode d'expression de celle-ci, notamment à travers sa modalité, peut être étudiée. Ainsi, l'interruption peut s'exprimer à travers différentes modalités (sonore, visuelle, tactile...) et pour chacune avoir des méthodes d'expression différentes (par exemple un bip ou la voix pour le son, la modulation de la transparence des messages visuels, l'usage de retours haptiques d'un volant de voiture lorsque le conducteur coupe la ligne blanche...). Elle peut être modulée par un critère temporel ou spatial (utilisation de sons 3D par exemple) et le choix de la méthode peut dépendre de l'importance de l'interruption (voir paragraphe précédent sur le « sens de l'interruption »).

Canal d'acheminement (*Channel of Conveyance*)

Cette catégorie décrit le canal de communication utilisé dans la tâche et comment l'interruption a été transmise : face-à-face, communication synchrone / asynchrone, directe ou indirecte. Dans les cockpits d'avion par exemple, la tâche principale visuelle peut être interrompue sur un canal différent (alerte sonore...).

Activité humaine impactée par l'interruption (*Human Activity Changed by Interruption*)

Cette catégorie va de pair avec la suivante et permet de prendre en compte l'activité de la personne interrompue. Plus précisément, les travaux cités par McFarlane et Latorella (2002) étudient la similarité entre la tâche interrompue et la nouvelle tâche, ainsi que la capacité de l'être humain à se préparer à l'interruption.

Effet de l'interruption (*Effect of Interruption*)

L'idée que les interruptions ont un impact (positif ou négatif) sur les performances de l'être humain interrompu est généralement admise. Cette catégorie se focalise sur l'étude de ces conséquences, l'effet qu'ont les interruptions sur

l'être humain : changement de focus attentionnel, stress induit, impact sur les performances, sur la vitesse de reprise de la première tâche, etc.

2.3.1.2 L'aspect temporel de l'interruption

Dans une étude antérieure à la définition de cette taxinomie, Latorella (1996) définit un modèle pour la gestion des interruptions, le IMSM (*Interruption Management Stage Model*). Ce modèle décrit la gestion des interruptions par l'être humain comme l'enchaînement de plusieurs étapes de traitement des informations (figure 2.4).

La tâche principale est composée de tâches discrètes, pouvant être décrites à travers une hiérarchie de buts. L'interruption, quant à elle, est décrite comme la succession d'un signal et de plusieurs étapes (détection du signal, interprétation, rejet ou intégration immédiat ou différé).

McFarlane et Latorella (2002) intègrent ce modèle dans leur réflexion et proposent un ensemble de recommandations pour la conception d'interface intégrant les 3 phases de l'interruption (avant, pendant, après) et les travaux correspondants aux huit catégories de la taxinomie. Ces recommandations portent principalement sur l'intégration des indications pré-interruption afin qu'elles soient en phase avec l'importance et/ou l'urgence de la tâche interruptive, sur la réduction des effets de l'interruption en faisant en sorte d'interrompre au meilleur moment (§ 2.3.3) ou la facilitation de la reprise post-interruption, à travers des indications sur les actions effectuées avant l'interruption. Les sections suivantes se focalisent également sur ces aspects relatifs aux interfaces homme-machine et précisent les recommandations émises par McFarlane & Latorella.

2.3.2 La psychologie de l'interruption

Dans le domaine de la psychologie cognitive, les effets de différents facteurs (préparation, longueur de l'interruption, entraînement, ...) sur l'interruption ont été étudiés à travers différentes approches : le contrôle exécutif (Monsell, 2003, voir aussi §2.2), la théorie de la mémoire prospective (Dodhia et Dismukes, 2003b,a), celle de la mémoire à long terme (Oulasvirta et Saariluoma, 2004, 2006) et la théorie de l'activation des buts en mémoire (*memory for goals*, Altmann et Trafton (2002)), appuyée par le modèle ACT-R (*Adaptive Control of Thought-Rational*, Anderson et Lebiere (1998)).

Les paradigmes expérimentaux et les métriques utilisées pour mesurer l'impact de l'interruption sur l'être humain s'apparentent aux travaux menés dans le domaine du *Task Switching* (section 2.2). Les tâches étudiées se situent à un niveau d'abstraction souvent plus élevé (sur l'échelle de Cooper et Shallice (2000)) par rapport aux

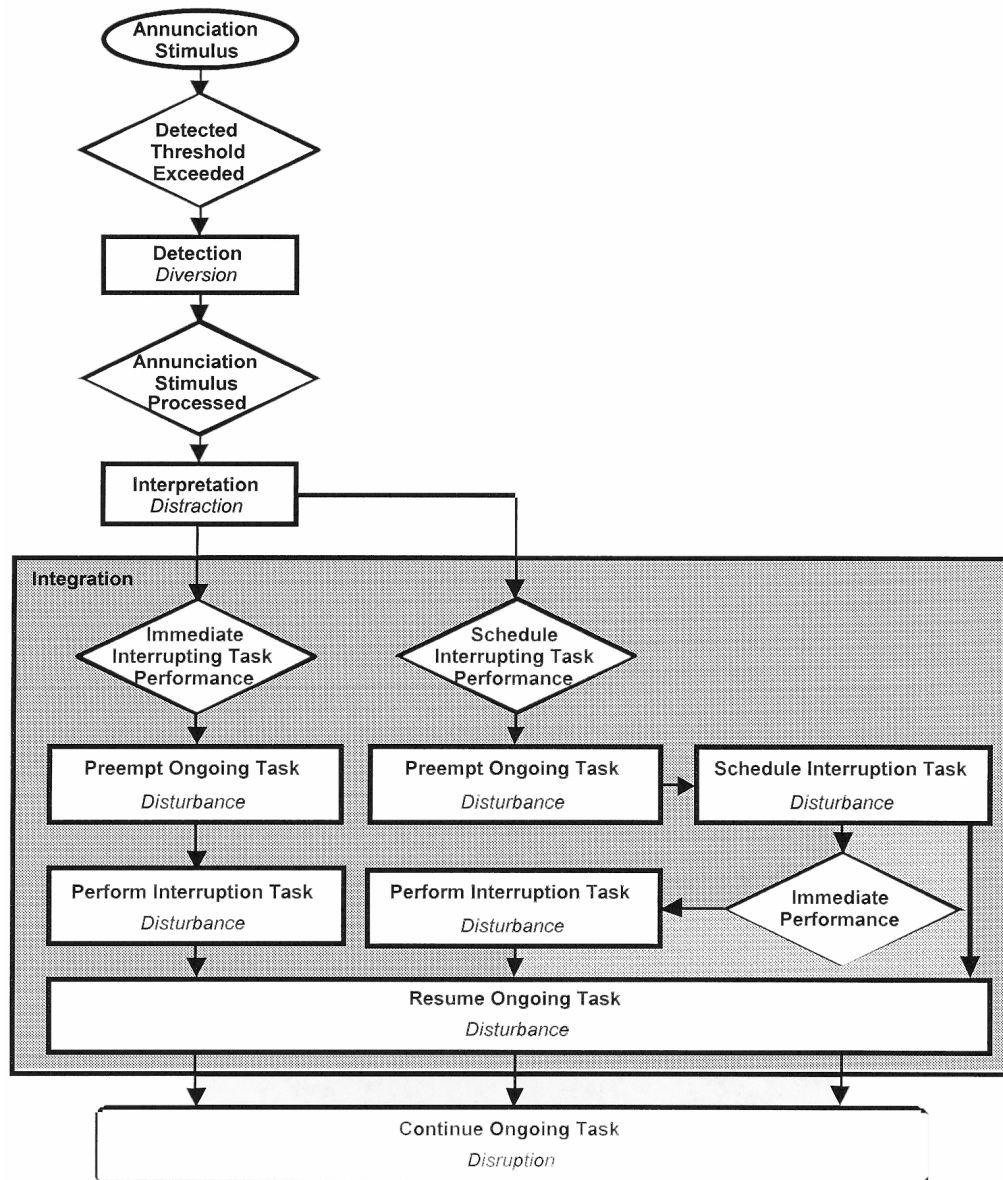


FIGURE 2.4 : le modèle de gestion des interruptions (*Interruption Management Stage-Model*) de Latorella (1996).

tâches « simples » présentées dans la section 2.2 ; la tâche principale consistant ici à réaliser une mission plus ou moins complexe à travers une application informatique.

La figure 2.5 présente la chronologie du déroulement d'une interruption selon les travaux de la psychologie cognitive (Trafton et Monk, 2007) : une tâche primaire est interrompue par une tâche secondaire, après l'occurrence d'une alerte pendant un délai plus ou moins long (« délai d'interruption »). Le temps de reprise (mesuré entre la fin de la tâche secondaire et la première action sur la tâche primaire) permet

de mesurer l'aspect perturbant de l'interruption.

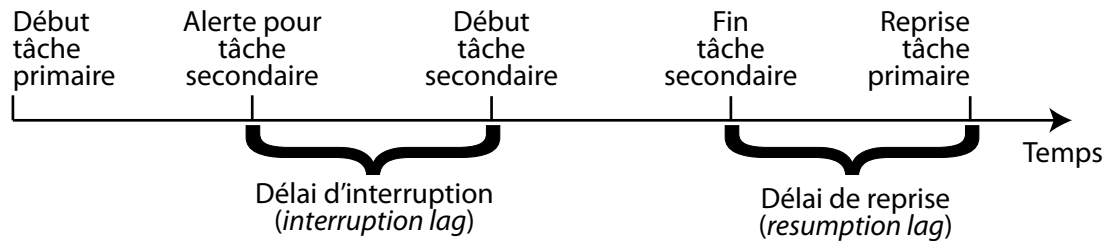


FIGURE 2.5 : chronologie d'une interruption (Trafton et Monk, 2007)

Premier résultat

A travers une expérimentation mettant en jeu une tâche complexe d'allocation de ressources (le sujet doit assigner des objectifs à un ensemble de tanks), interrompue par une tâche secondaire de classifications d'objets (durant entre 30 et 45 secondes), Altmann et Trafton (2004) ont montré que le simple fait d'interrompre le sujet avait un impact sur le temps d'exécution des instructions. Le délai de reprise après une interruption était le double (3,8 secondes) par rapport au temps moyen entre deux actions de la tâche primaire (1,9 seconde) (Altmann et Trafton, 2004, expérience 1).

2.3.2.1 Les facteurs impactant l'interruption

L'interruption a ainsi un effet négatif sur la réalisation de la tâche primaire de l'être humain. Nous présentons une liste des facteurs ayant fait l'objet de travaux et dont l'impact sur l'interruption, et notamment sur la reprise de la tâche primaire (à travers la mesure du délai de reprise), a pu être montré expérimentalement. Nous nous sommes en partie inspirés de l'état de l'art de Trafton et Monk (2007) pour établir cette liste.

Longueur de l'interruption

Monk *et al.* (2008); Hodgetts et Jones (2006) montrent dans leurs études respectives que la longueur de l'interruption a un effet sur la reprise de la tâche principale : plus l'interruption est longue, plus celle-ci est perturbante pour la reprise. Ces résultats contredisent ceux de Gillie et Broadbent (1989) qui n'avaient pas trouvé d'effet significatif de la longueur de l'interruption sur la reprise. Trafton et Monk (2007) justifient cette différence par la différence de méthodologie employée, indiquant que les interruptions de Gillie et Broadbent (1989) étaient trop longues et les métriques utilisées ne permettaient pas de conclure (Gillie et Broadbent (1989) mesuraient le taux d'erreur et le temps de réalisation de la tâche dans leur globalité).

Préparation

Le facteur de préparation consiste à laisser du temps au sujet pour se préparer durant l'intervalle d'interruption (entre l'alerte et le début de la tâche secondaire, voir figure 2.5). Ainsi, Trafton *et al.* (2003) montrent que lorsque l'opportunité est donnée au sujet de se préparer (délai d'interruption de huit secondes), celui-ci obtient une meilleure performance lors de la reprise de la tâche primaire par rapport aux sujets n'ayant pas eu de préparation avant l'interruption (délai d'interruption nul). Les résultats montrent également qu'avec de l'entraînement, les sujets n'ayant pas de temps de préparation améliorent leurs temps de reprise : sur la dernière session d'expérimentation, le temps de reprise est très proche pour les deux groupes.

Ces résultats sont en phase avec ceux de Dodhia et Dismukes (2003b,a) qui trouvent une différence significative sur leur essais lorsque le sujet a l'opportunité de se préparer pendant quatre secondes par rapport à la condition sans préparation.

Rôle des indications lors de la reprise

Afin de faciliter la reprise de la tâche primaire et de « réactiver en mémoire » l'objectif associé, une indication permettant au sujet de se remémorer sa dernière action avant l'interruption peut lui être fournie. Les résultats des études sur ce sujet montrent que la reprise de la tâche primaire est facilitée lorsqu'une indication flagrante (flèche rouge géante, texte clignotant...) est fournie (Trafton *et al.*, 2005) tandis qu'une indication plus subtile (curseur sur l'emplacement de la dernière action par exemple) n'avait pas d'effet par rapport à la condition sans indication (Trafton *et al.*, 2005; Czerwinski *et al.*, 2000). Trafton et Monk (2007) proposent un exemple de rappel avec une mémoire des trois dernières actions réalisées avant l'interruption, visualisées à travers trois flèches de couleur rouge dégradée (la flèche plus foncée correspond à l'action la plus récente et inversement). Nous reproduisons cet exemple sur la figure 2.6.

Similarité entre les tâches

L'étude de la similarité entre la tâche primaire et la tâche secondaire ne donnent pas les mêmes résultats. Dodhia et Dismukes (2003a) ne trouvent aucun impact significatif sur l'interruption que les deux tâches soient similaires ou non.

Au contraire, Gillie et Broadbent (1989, expérimentation 3) montrent que la présence d'une tâche secondaire similaire à la tâche primaire (rappel de listes de catégories) perturbe davantage l'interruption que pour deux tâches distinctes (rappel de liste et tâche arithmétique). En se fondant sur le même paradigme expérimental, Edwards et Gronlund (1998) confirment ce résultat. De la même manière, Oulasvirta et Saariluoma (2004, expérimentation 3) ont montré que des messages interruptifs sémantiquement plus proches de la tâche primaire

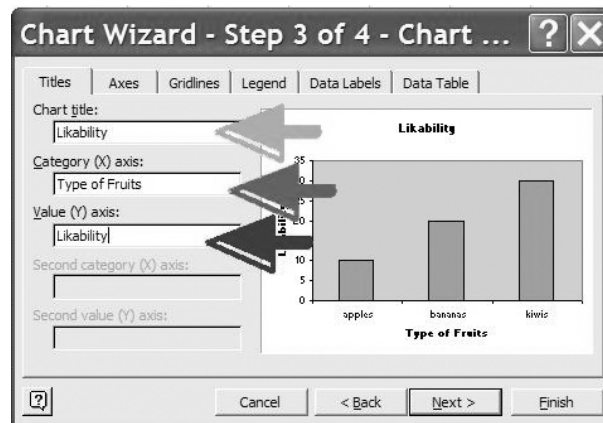


FIGURE 2.6 : exemple de trois indications associées aux trois dernières actions effectuées avant l'interruption. Le dégradé (initialement du rouge foncé au rouge clair) indique l'ordre des actions (Trafton et Monk, 2007).

étaient plus perturbants.

Enfin, Czerwinski *et al.* (1991) rapportent que la similarité des tâches d'un point de vue de l'affichage a un effet perturbant sur l'interruption, allant dans le sens des résultats d'interférence cognitive présentés dans le paragraphe 2.2.2.1. Ainsi, la similarité entre les deux tâches semble avoir un impact sur l'interruption mais les résultats divergents laissent à penser que ces effets dépendent des tâches en jeu et des informations associées à ces tâches.

Influence de la complexité de la tâche primaire

Lorsque l'on étudie la complexité de la tâche primaire, les travaux de Speier *et al.* (2003) indiquent que les tâches plus complexes sont perturbées par les interruptions tandis que, de façon contre-intuitive peut-être, la performance sur des tâches simples peut être améliorée. Ils suggèrent que cela peut provenir du fait que, pour des tâches considérées comme « trop simples » ou « ennuyeuses » par l'être humain, les interruptions permettent de l'aider à re-focaliser son attention sur ces tâches et ainsi améliorer ses performances. Cet aspect positif de l'interruption sur des tâches simples a été également noté par Zijlstra *et al.* (1999), où les sujets adaptaient leurs stratégies de travail en fonction des interruptions (mais ces stratégies évoluent également en fonction de l'expertise des participants et de leur habitude à être interrompus).

Influence de la complexité de la tâche secondaire

Dans leurs expérimentations, Gillie et Broadbent (1989); Hodgetts et Jones (2005); Cades *et al.* (2007) ont montré que la complexité de la tâche secondaire, définie comme l'effort mental nécessaire à sa réalisation, influence de manière négative les performances de l'être humain sur la tâche primaire interrompue. Dans leur étude, Hodgetts et Jones (2005) indiquent également que plus l'interruption a lieu tôt dans la réalisation de la tâche primaire (une adap-

tation du problème des tours de Hanoï), plus celle-ci était perturbatrice. Cades *et al.* (2007) suggèrent également que les mauvais résultats liés aux tâches plus complexes sont liés au fait qu'une tâche secondaire complexe, lorsqu'elle interrompt la tâche primaire, laisse moins de temps au sujet pour stocker en mémoire les informations sur la tâche primaire (stockage qui serait effectué sur le temps de réalisation de la tâche secondaire).

Influence de l'entraînement

Dans (Trafton *et al.*, 2005), les auteurs ont remarqué qu'au fur et à mesure de la réalisation des expérimentations, l'écart de performance entre le groupe sans indication pré-interruption et celui qui était prévenu se réduisait, laissant à supposer qu'avec l'expérience, l'impact des interruptions est de moins en moins important sur la performance des sujets. Un résultat semblable avait été relevé dans (Trafton *et al.*, 2003). Cades *et al.* (2006) ont dédié une expérimentation à ce sujet, montrant que les sujets entraînés à être plus souvent interrompus étaient moins gênés par les interruptions, après un temps équivalent de pratique de la tâche primaire. Ce résultat suggère que l'entraînement des sujets, et notamment l'intégration d'interruptions dans les scénarios d'entraînement des opérateurs de systèmes « interruptibles » permettra de réduire l'effet perturbateur de ces interruptions.

2.3.2.2 Propositions de solutions basées sur ces facteurs

Boehm-Davis et Remington (2009) proposent une revue de solution de réduction de l'effet « perturbateur » des interruptions, basée sur l'étude des processus cognitifs en jeu. Les correctifs proposés par les auteurs sont fonction de deux paramètres (figure 2.7) : le coût de l'erreur induite par l'interruption et la charge cognitive associée à la tâche interrompue.

- Plus la charge cognitive augmente, plus il devient important de proposer des solutions d'interruptions à faible impact cognitif.
- De la même manière, plus le coût d'une erreur induite par l'interruption est important, plus la solution proposée doit être fiable (peu de fausses alertes et grande précision) et efficace. Lorsque le coût de l'erreur est faible (à gauche des pointillés sur la figure 2.7), le fait de manquer une alerte ou de ne pas reprendre correctement la tâche primaire n'a pas de conséquence « catastrophique » (par exemple, dans un environnement de bureau). A l'inverse, une alerte manquée dans la région où le coût de l'erreur est très important peut impliquer des conséquences très graves, allant jusqu'à la perte de vie humaine (dans un environnement médical par exemple).

La stratégie proposée pour la mise en œuvre de ces solutions est la suivante :

- Lorsque le coût des erreurs induites par l'interruption est faible (région de gauche dans la figure 2.7), le système génère l'interruption en fonction de son importance, priorité et pertinence et en fonction de la charge de l'opérateur (voir la section 2.3.3 pour des exemples de moyens de coordination) ;

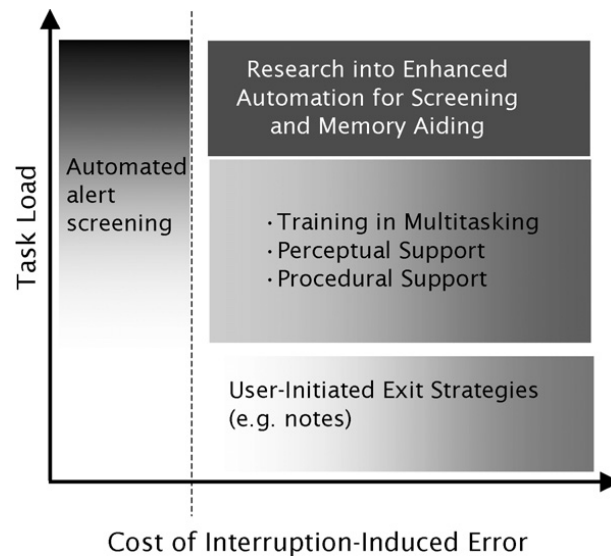


FIGURE 2.7 : propositions de solutions d'interruption en fonction du coût des erreurs et de la charge induite(Boehm-Davis et Remington, 2009).

- Lorsque le coût des erreurs induites est plus important, des stratégies d'aide à la reprise peuvent être mises en œuvre :
 - si la charge cognitive est faible, l'utilisateur gère lui-même sa stratégie de sortie/reprise de la tâche principale, le système peut notamment lui laisser du temps pour se préparer et mémoriser l'état de la tâche primaire avant l'interruption ;
 - un support à la perception (indices) et/ou à la reprise des procédures peut être apporté, ainsi qu'un entraînement à la gestion de contextes multitâches ;
 - dans les systèmes où les risques sont très élevés en cas d'erreur et où la charge de l'opérateur est importante (aviation, contrôle aérien...), l'automatisme peut prendre le relais sur les tâches de l'être humain afin de lui faciliter la tâche. Ces solutions font intervenir d'autres domaines de recherches spécifiques (comme par exemple l'allocation dynamique de tâche).

2.3.3 Coordination de l'interruption : comment interrompre et à quel moment ?

Dans la taxinomie de McFarlane (1997), une catégorie est dédiée aux méthodes de coordination de l'interruption entre l'humain et la source de l'interruption. L'application de ces méthodes de coordination est pertinente lorsque l'interruption est exogène (déclenchée par un élément extérieur). Dans le domaine de l'interaction homme-machine, cette coordination se fait généralement entre un ordinateur et l'être humain opérant le système, dans le cas où la machine doit interrompre l'opérateur. Nous présentons dans cette section différentes études dont l'objet est de déterminer le meilleur moment pour interrompre la tâche primaire afin de minimiser l'impact

sur les performances de l'être humain.

2.3.3.1 Comparaisons des méthodes de coordination de la taxinomie de McFarlane

McFarlane (2002) a mené une étude permettant de comparer les quatre méthodes de coordination identifiées dans sa taxinomie : *l'interruption immédiate* (pas de coordination), *l'interruption négociée* (l'utilisateur choisit s'il accepte l'interruption), *l'interruption calculée* (le système calcule selon un algorithme prédéfini si l'utilisateur peut être interrompu) et *l'interruption programmée* (toutes les x secondes). La tâche primaire de l'expérimentation est une adaptation d'un jeu vidéo de NintendoTM appelé « *Fire* » dans lequel le joueur doit diriger deux brancardiers militaires réceptionnant des diplomates sautant d'une ambassade américaine en feu (figure 2.8a). Les diplomates doivent rebondir trois fois avant d'être complètement réceptionnés et considérés comme sains et saufs. La difficulté de la tâche est proportionnelle au nombre de diplomates présents simultanément à l'écran, celle-ci est contrôlée afin de s'assurer que les participants ne se découragent pas. La tâche secondaire, conçue pour être très différente de la tâche primaire afin d'éliminer les effets de similarité (§ 2.3.2.1), est inspirée des travaux de Stroop (1935). Elle consiste à comparer et sélectionner les objets de même forme ou de même couleur (figure 2.8b).

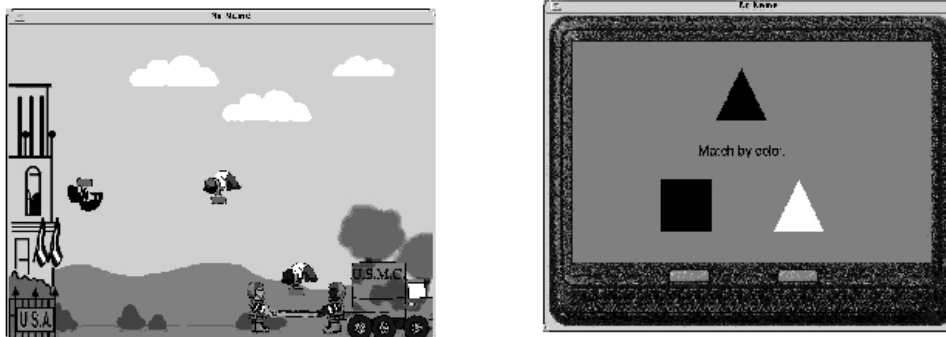


FIGURE 2.8 : captures d'écran de (a) la tâche primaire et de (b) la tâche secondaire de l'expérimentation de McFarlane (2002).

Six configurations différentes sont testées durant cette expérimentation :

- Deux configurations correspondent aux conditions de « contrôle » : le participant doit réaliser uniquement l'une des tâches (primaire ou secondaire) ;
- Dans la condition d'*interruption immédiate*, la tâche secondaire est affichée sans préavis et le participant doit la réaliser directement avant de reprendre la tâche primaire ;
- Dans la condition d'*interruption négociée*, la tâche secondaire est affichée pendant 150 millisecondes avant de disparaître. Le sujet peut alors choisir quand

il veut réaliser la tâche secondaire, en appuyant sur une des touches du clavier associée à cette tâche, afin de la ramener au premier plan. Dans cette configuration, plusieurs tâches secondaires peuvent être mises en attente, le sujet peut alors choisir de les réaliser quand il veut (et revenir à la tâche primaire à loisir) ;

- Dans la condition d'*interruption calculée*, le système estime la charge de travail du participant en fonction du nombre de diplomates présents à l'écran. Lorsqu'il estime que celle-ci est basse, il interrompt le sujet sans notification et celui-ci doit réaliser la tâche secondaire. L'algorithme utilisé s'assure qu'il y a moins de six interruptions (six tâches secondaires) dans la file d'attente (notamment, quelle que soit la charge de travail estimée, une septième interruption entrant dans la file déclenchera l'exécution de la première).
- Dans la condition d'*interruption programmée*, le système met les interruptions en file d'attente automatiquement et les impose au participant toutes les 25 secondes, sans notification.

Les mesures de performance des sujets à travers les différentes configurations (taux de sauvetage des diplomates, taux d'erreur sur la tâche secondaire, temps de réponse) montre que les meilleurs résultats ont été obtenus dans la condition d'*interruption négociée* suivie de l'*interruption immédiate*. A travers les questionnaires post-expérimentaux, les participants ont également indiqué préférer la condition d'*interruption négociée*. A partir des données collectées (quantitatives et qualitatives), McFarlane propose la recommandation générale suivante pour le design d'interface :

“

L'interruption négociée est la meilleure solution sauf pour les problèmes où de légers écarts temporels sur le début de la tâche secondaire sont critiques. Dans ce cas, l'interruption immédiate est la meilleure.

”

Enfin, McFarlane (2002) propose un classement des différentes solutions en fonction des objectifs de conception. Le tableau 2.2 détaille ces recommandations. L'auteur précise que ces recommandations sont issues uniquement des résultats (quantitatifs et qualitatifs) et ne peuvent donc être généralisées qu'avec prudence.

S'appuyant sur les travaux de McFarlane (2002), Franke *et al.* (2002) proposent une variante des stratégies de coordination dans laquelle l'importance de l'interruption est évaluée dynamiquement (en fonction de l'importance accordée à la tâche primaire et à la tâche secondaire). Une stratégie est alors sélectionnée par le système en fonction de ces données (Franke *et al.*, 2002, tableau 1) : l'interruption peut être immédiate, négociée (le système propose alors une action par défaut selon l'importance : « interrompre maintenant ? » ou « reporter l'interruption ? ») ou reportée à

TABLEAU 2.2 : recommandations sur le choix du mode de coordination, sélectionné en fonction des objectifs de conception (McFarlane, 2002).

Design Goal	Best	Worst
Accuracy on continuous task	Negotiated	Scheduled
Efficiency on continuous task	Negotiated/Mediated	Immediate/Scheduled
Fewest task switches	Scheduled	Immediate
Accuracy on intermittent task	Not Immediate	Immediate
Completeness on intermittent task	Immediate/Mediated	Scheduled/Negotiated
Promptness on intermittent task	Immediate	Scheduled/Negotiated
Efficiency on intermittent task	Negotiated/Scheduled	Immediate
Keying accuracy	Negotiated/Scheduled	Immediate
User preference	Negotiated/Mediated	Immediate/Scheduled
User perception of their own accuracy on continuous task	Not Immediate	Not Negotiated
User perception of least interruptive	Negotiated/Mediated	Immediate/Scheduled
User perception of most predictable	Scheduled/Negotiated	Immediate/Mediated
User perception of complexity of continuous task when interrupted	Negotiated/Mediated	Immediate/Scheduled

la fin de la conversation courante. Le système étudié est un cockpit d'avion où les échanges avec le système se font par la voix. Le système joue notamment sur cette modalité sonore en changeant la voix de l'automate annonçant la tâche secondaire (par rapport à la voix utilisée pour les dialogues de la tâche principale).

2.3.3.2 Modèle bayésien de calcul d'interruption

L'approche de Gievska et Sibert (2004) repose sur une méthode d'*interruption calculée* à partir de l'inférence de l'activité de l'utilisateur et basée sur la modélisation des interruptions à l'aide de réseaux bayésiens (BBN : *Bayesian Belief Networks*). Les auteurs (voir aussi Gievska et Sibert, 2005; Gievska *et al.*, 2005) proposent ainsi de décrire les interruptions selon une taxinomie (figure 2.9) à trois axes : contexte de la tâche, contexte de l'utilisateur et contexte de l'environnement.

Les différents attributs de cette taxinomie sont ensuite encodés dans un réseau de croyance bayésien, représentant les relations causales entre ces différents concepts. Avec ce modèle, les auteurs souhaitent « améliorer l'intelligence de la machine à l'aide d'informations sur le contexte de l'interruption » plutôt que de « construire un modèle de la psychologie de l'être humain lors des interruptions » (Gievska et Sibert, 2004, page 420). Le déploiement du *framework* se déroule en trois étapes :

1. Sélection, dans la taxinomie, des variables contextuelles pertinentes pour le domaine étudié ;

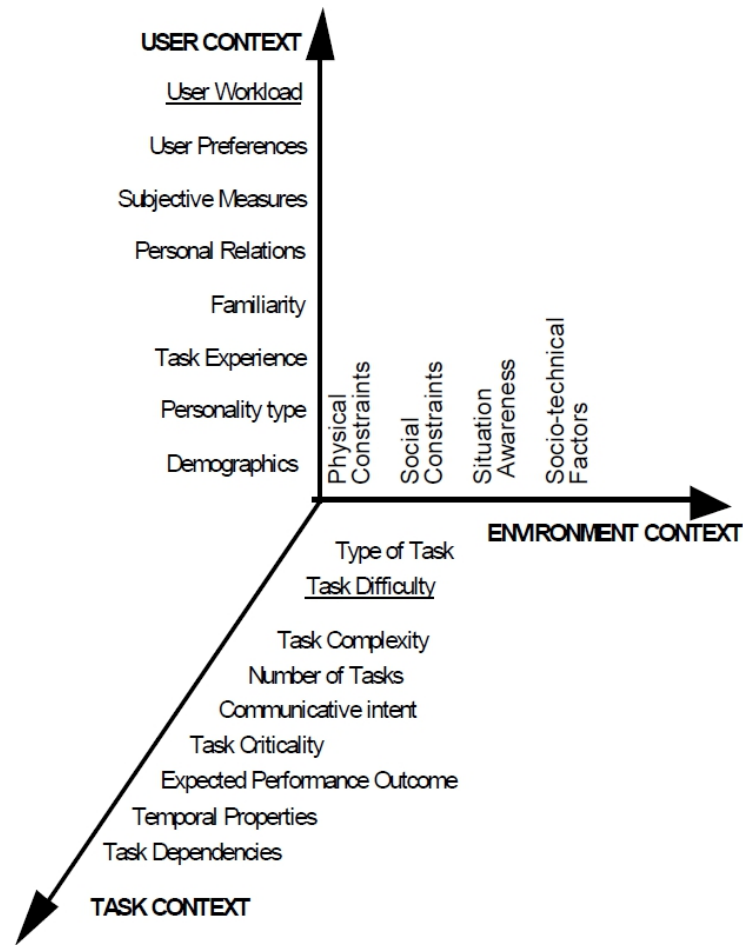


FIGURE 2.9 : taxinomie des interruptions proposées par Gievska et Sibert (2004).

2. Identification des sources d'information nécessaires pour connaître l'état de ces variables ;
3. Construction et entraînement (*offline*) du réseau bayésien : création des liens entre les variables et spécification des probabilités pour chacun de ces liens.

Pour mener à bien ces étapes, les auteurs ont mené une étude pilote sur quinze participants et ont parcouru la littérature relative à leur domaine expérimental (l'expérimentation a lieu sur un simulateur d'allocation de ressources tactiques et d'évaluation tactique pour l'armée américaine). Le réseau correspondant à cette étude est décrit à un niveau élevé dans la figure 2.10, chaque variable se décompose en sous-réseaux bayésien, que les auteurs ne présentent pas dans leurs publications.

Après avoir « entraîné » le réseau bayésien, celui-ci a été implémenté au sein du logiciel et utilisé afin de calculer le meilleur moment pour les interruptions. Lors des expérimentations, les auteurs montrent que lorsque l'interruption est calculée par le système, les résultats (en termes de temps de reprise et de taux d'erreur sur les tâches primaire et secondaire) sont meilleurs par rapport aux interruptions

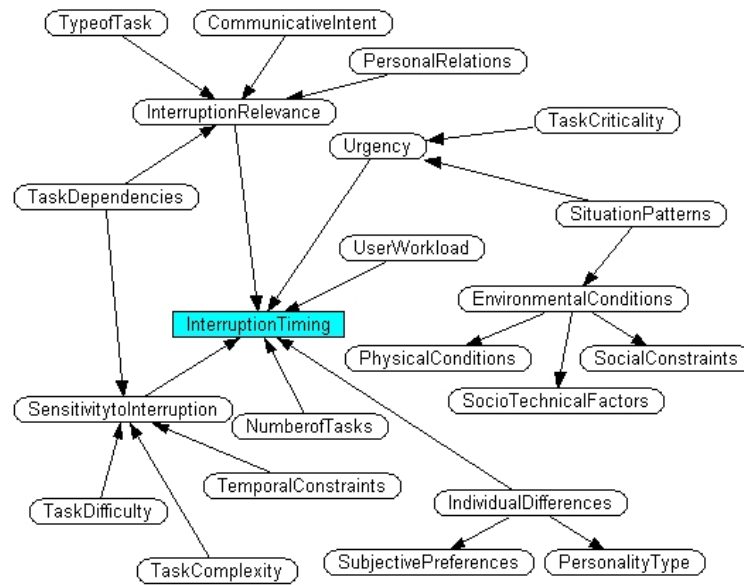


FIGURE 2.10 : dépendances « haut niveau » entre les variables de la taxinomie et leur influence sur le timing de l'interruption (extrait de Gievska *et al.* (2005)).

immédiates (Gievska et Sibert, 2004). Une analyse des mesures subjectives (réponses aux questionnaires post-expérimentation) montre que les participants perçoivent l'interruption calculée comme un moyen de diminuer les sentiments de distraction, de frustration et d'agacement mais ont l'impression d'avoir une charge de travail supplémentaire par rapport aux interruptions immédiates (Gievska *et al.*, 2005).

2.3.3.3 Intégration des interruptions calculées à une méthodologie de conception

Godbole et Smari (2006a) proposent une méthodologie de conception intégrant un modèle d'interruption similaire. L'approche présentée se décompose en trois phases (figure 2.11) :

- Un processus de collecte de données centré sur l'utilisateur (entretiens et questionnaires, observations sur le terrain, analyse de la littérature) permet de recueillir les informations sur leur activité ainsi que les préférences et habitudes en termes d'interruption au sein de cette activité.
- Les données collectées sont ensuite converties en règles et contraintes par le concepteur du système afin de « nourrir » un modèle d'apprentissage. Elles serviront également d'ensemble d'apprentissage pour le modèle.
- Une phase de collecte des préférences des utilisateurs sur le système est menée lors de son utilisation et permet d'affiner les règles et contraintes du modèle afin qu'il soit adapté au mieux aux situations de travail des utilisateurs.

A la différence des travaux de Gievska et Sibert (2004), la capacité de l'utilisateur à être interrompu est définie ici à partir des informations système et ne se

2.3. LES INTERRUPTIONS EN INTERACTION HOMME-MACHINE

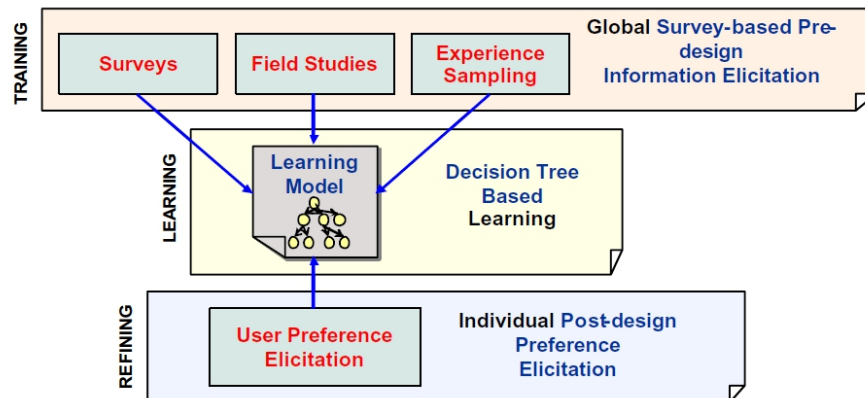


FIGURE 2.11 : trois phases de la mise en place du modèle d'apprentissage dans la méthodologie de Godbole et Smari (2006a).

repose pas sur l'inférence de l'activité de l'utilisateur. Dans un exemple d'application (Godbole et Smari, 2006b), cette méthodologie est appliquée à la conception d'un système de gestion des interruptions pour téléphone mobile. Les auteurs précisent également qu'elle peut s'intégrer aux méthodologies de conception centrée-utilisateur existantes, profitant des étapes d'échanges avec les utilisateurs pour construire, valider et raffiner le modèle d'apprentissage (Godbole et Smari, 2006a, figure 5).

2.3.3.4 Détection automatique des « creux » de charge de travail

Dans leurs travaux, Bailey et ses collègues (Adamczyk et Bailey, 2004; Iqbal et Bailey, 2005; Bailey et Konstan, 2006) étudient l'impact de la charge de travail sur l'interruption à partir d'un modèle hiérarchique des tâches de l'utilisateur.

Dans leur modèle, chaque tâche (comme la lecture de courrier électronique) est divisée en sous-tâches « grossières » (*ouvrir la boîte de réception* dans un logiciel de messagerie par exemple), elles-mêmes divisées en tâches « fines » (*ouvrir un message, sélectionner du texte*). Dans une première expérimentation, Adamczyk et Bailey (2004) ont su prédire quels étaient les meilleurs et pires instants pour interrompre la tâche en cours, en fonction de la charge de travail de l'utilisateur, inférée à partir de la décomposition des tâches (figure 2.12). Les participants, conformément à l'hypothèse des auteurs, ont été moins gênés lorsque l'interruption avait lieu entre deux tâches « grossières ». À l'inverse, l'interruption pendant une tâche « fine » (moment où la charge de travail est considérée par les auteurs comme élevée) avait un impact négatif. Ces résultats ont été reproduits à plusieurs reprises (Bailey et Konstan, 2006; Bailey et Iqbal, 2008).

Les travaux de cette équipe de Microsoft Research ont mené au développement du *framework* OASIS (*Omniscient Automated System for Interruption Scheduling*, Iqbal et Bailey, 2010), qui permet de distribuer automatiquement les notifications des applications du système en fonction de leur importance et de l'activité de l'uti-

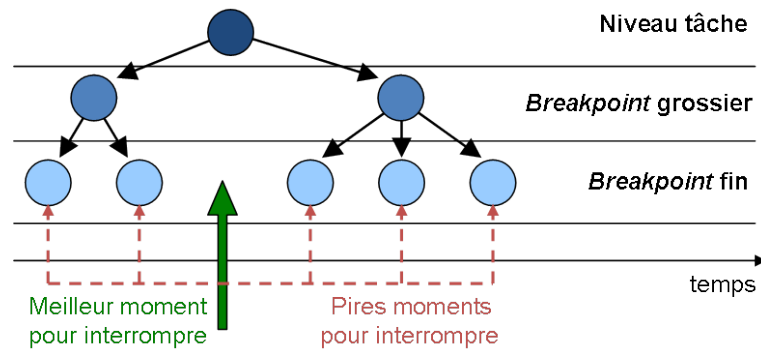


FIGURE 2.12 : hiérarchie de décomposition des tâches (adapté de Adamczyk et Bailey (2004)).

lisateur. L'intégration d'OASIS dans l'interaction entre l'homme et la machine est présentée en figure 2.13.

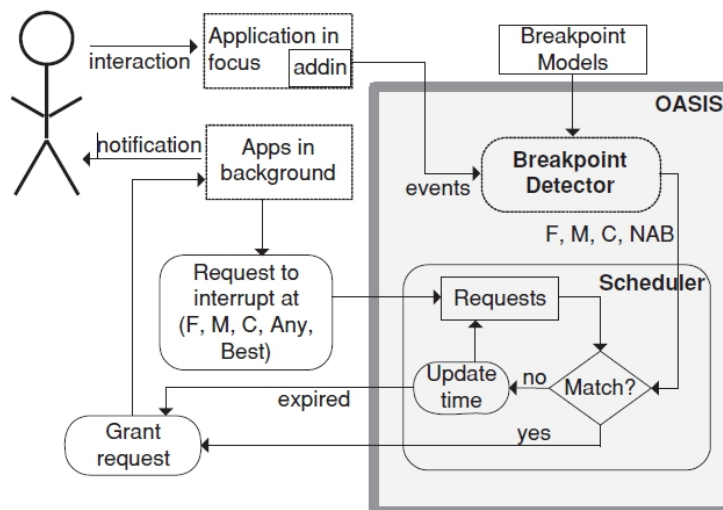


FIGURE 2.13 : intégration du framework OASIS et de ses opérations (extrait de Iqbal et Bailey, 2010).

Deux types d'événements sont gérés : les événements systèmes (en particulier ceux relatifs à la souris, au clavier et à la gestion des fenêtres du système d'exploitation, en l'occurrence Microsoft Windows) et les événements propres aux applications liées (par exemple, dans (Iqbal et Bailey, 2010), les événements générés par MS Visio tels que la création de nouveaux objets ou l'édition de leur contenu). A partir de modèles statistiques (*breakpoints models*) pré-chargés dans le système, OASIS convertit en temps réel l'enchaînement des événements en *breakpoints* définis selon 3 niveaux : « grossier », « moyen » et « fin », correspondant aux niveaux hiérarchiques

de la décomposition des tâches (figure 2.12).

Lorsqu'une application doit notifier l'utilisateur, l'information est envoyée à OASIS en précisant son importance (« pertinent » ou « urgent ») ainsi que la préférence de distribution (« le moins perturbant » ou « le plus rapide »); ces paramètres définissent le niveau du prochain *breakpoint* pour lequel la notification sera affichée. Ainsi, le *framework* OASIS permet de distribuer la notification au meilleur moment en fonction des besoins de l'application tout en respectant l'utilisateur.

Les travaux présentés sont restreints actuellement au cas des notifications (qui, bien que n'étant pas des tâches peuvent induire une interruption, qui serait alors négociée) mais nous pouvons imaginer que ce principe puisse être étendu au cas général des interruptions.

2

2.3.4 Les interruptions dans les modèles

2.3.4.1 La notion de « singularité » de Jambon

Dans le domaine de l'interaction homme-machine, Jambon (1996) propose d'étendre la notion d'interruption en introduisant celle de « singularité » qui « englobe toutes les tâches qui “singularisent” la tâche en cours, en quelque sorte tout ce qui s'éloigne de la trajectoire d'interaction normale sans erreurs ni interruptions d'aucune sorte » (Jambon, 1996).

A partir de cette notion de singularité, basée sur les relations temporelles (séquence avec ou sans reprise, parallélisme, entrelacement), il propose d'étendre le pouvoir d'expression de trois formalismes de description des tâches, chacun adapté à l'une des phases du cycle de conception.

- La notation MAD (Méthode Analytique de Description) de Scapin et Pierret-Golbreich (1989) pour l'analyse des tâches.
- La notation UAN (*User Action Notation*) de Hartson et Gray (1992) pour la description de l'interface homme-machine.
- Les réseaux de Pétri (Peterson, 1981) pour la spécification logicielle résultante.

Jambon énonce également des principes de traduction des extensions proposées, afin de créer un lien entre ces trois formalismes et définit des tableaux d'équivalence utilisables pour passer d'une notation à l'autre. Ces travaux sont illustrés par une application au cas du Publiphone (le téléphone public à carte, que l'on trouve notamment dans les « cabines téléphoniques »). Notons que ces principes de traductions ne sont pas nécessairement valides dans les deux sens, certains formalismes étant plus détaillés que d'autres. Enfin, leur diffusion est limitée par l'absence d'outils spécifiques soutenant la traduction d'un formalisme vers le suivant dans le cycle de conception.

2.3.4.2 Évaluation de la tolérance des systèmes aux interruptions

A la différence des approches que nous avons présentées dans les sections précédentes, qui visent à améliorer (ou à minimiser) l'impact de l'interruption sur l'être humain, Palanque *et al.* (2009b) proposent une méthode formelle d'évaluation de la tolérance des systèmes aux interruptions. Cette approche consiste à décrire :

- les tâches de l'utilisateur (avec le formalisme *ConcurTaskTrees* de Paternò *et al.* (1997)),
- le système permettant la réalisation de ces tâches décrites (formalisme ICO, *Interactive Cooperative Objects*, Palanque et Bastide (1994)),
- les interruptions potentielles (à travers le formalisme ICO également)

puis à injecter des valeurs estimant les performances cognitives et motrices liées à chaque interaction et tâches système. Ces valeurs peuvent être issues de mesures expérimentales ou de la littérature (loi de Fitts, théories de la cognition, modèles de type KLM...). Cette méthode permet d'évaluer l'impact de la fréquence des interruptions sur les performances prédites de l'utilisateur et de comparer les performances de plusieurs systèmes associés à la même tâche en fonction des interruptions envisagées.

L'exemple produit par Palanque *et al.* (2009b) présente deux moyens d'interaction pour la suppression de fichier dans une application simple : le *drag & drop* (interaction à la souris) et le *speak & drop* (interaction souris & voix) et compare les performances attendues en fonction de la fréquence d'interruption d'une tâche simple (apparition d'une fenêtre modale avec un seul bouton *OK*). L'évaluation estime que le *drag & drop* est plus efficace pour une fréquence d'interruption inférieure à une par minute (et inversement). Pour les deux moyens d'interaction, la performance estimée décroît lorsque la fréquence d'interruption augmente.

A travers cette approche formelle, Palanque *et al.* (2009b) proposent un outil d'évaluation de l'effet des interruptions sur les systèmes. Dans cette approche, les interruptions sont considérées comme des « système indépendants », dont l'impact est mesuré à travers l'injection de valeurs théoriques. L'aspect cognitif (modèle du processeur humain) est pris en compte dans l'évaluation du coût de la transition (Palanque *et al.*, 2009a) qui est vue comme une tâche distincte dont la valuation est bien définie (temps cognitif de perception + exécution de la tâche + temps de reprise de la tâche primaire). Nous notons, comme les auteurs, que l'évaluation des performances d'un système unique est soumise au bon choix des valeurs injectées tandis que la comparaison des performances de deux systèmes soumis aux interruptions et avec les mêmes valeurs injectées produit des résultats relatifs donc comparables.

2.3.5 Commentaire sur les Interruptions

La taxinomie présentée en introduction de cette section nous a permis de décrire les différents axes de travail relatifs aux interruptions de tâches en Interaction

Homme-Machine. Les travaux rapportés se focalisent sur certains de ces axes.

- La section 2.3.2.1, portant sur les aspects cognitifs des interruptions, étudie ainsi les effets de l'interruption et prend en compte les activités impactées par l'interruption (à travers l'étude de la similarité entre tâches notamment).
- La section 2.3.3 nous donne des éléments de réponse, qu'il s'agisse de recommandations ou de solutions méthodologiques, à propos des méthodes de coordination entre l'homme et la machine afin de choisir le moment de l'interruption.
- Enfin, nous avons présenté une méthodologie permettant d'évaluer l'effet des interruptions sur les performances d'un système en fonction des solutions proposées. Cette méthodologie répond en partie aux attentes de l'axe « méthode d'expression » et prend en compte les caractéristiques individuelles de la personne interrompue en injectant des valeurs dépendant notamment de l'expertise de l'utilisateur.

2.4 SYNTHÈSE DU CHAPITRE 2

Dans ce premier chapitre d'état de l'art, nous avons présenté trois approches pour l'étude des transitions entre tâches :

- La première partie (section 2.1) décrit l'approche par la modélisation des tâches, approche centrée sur l'activité des utilisateurs. Les modèles exposés intègrent tous la modélisation des relations entre les tâches avec plus ou moins de détails, allant d'un formalisme basé sur des standards (la notation CTT s'appuie sur les opérateurs normalisés LOTOS (ISO, 1989)) à des descriptions textuelles de ces transitions, laissant au concepteur la liberté de définir ces relations, ce qui rend difficile leur homogénéisation dans les modèles produits.
- Dans la deuxième partie (section 2.2), nous avons présenté les travaux sur les effets cognitifs des changements de tâche. Ces travaux de laboratoire décrivent différents paradigmes de *Task Switching* et les effets d'interférence en découlant, dépendant par exemple de la similarité des stimuli présentés, ou encore de la préparation du sujet au changement de tâches.
- La dernière partie de ce chapitre (section 2.3) présente enfin le cas particulier des interruptions entre tâches, leur influence sur la réalisation des tâches par les utilisateurs (aspect cognitif) ainsi que différentes méthodologies de conception ou d'évaluation intégrant les interruptions.

Nous remarquons que, même si les travaux présentés dans ce chapitre intègrent l'utilisateur final dans leurs considérations, les éléments d'information décrivant la situation attachée aux tâches étudiées sont très peu mis en avant, lorsqu'ils le sont.

Certains modèles de tâches présentés (CTT ou K-MAD par exemple) permettent de spécifier les objets manipulés par l'utilisateur dans son activité, mais aucun des travaux présentés n'intègre une description précise et formalisée des informations nécessaires à l'utilisateur pour la réalisation des tâches considérées. Le bilan « informationnel » des transitions n'est pas pris en compte de manière systématique lors de l'étude de ces changements de tâche, ne permettant pas d'étudier leur impact sur

la compréhension globale de la situation par l'utilisateur final.

Nous proposons ainsi dans le chapitre suivant d'aborder cette problématique du point de la situation, en présentant les travaux portant sur la description des éléments nécessaire à la compréhension de la situation, en particulier dans le cas de systèmes dynamiques complexes considéré dans notre étude.

3

Modéliser le contexte des transitions

Les systèmes complexes dynamiques que nous considérons dans cette thèse amènent leurs opérateurs à gérer un ensemble de tâches en constante évolution. Le contexte changeant de la mission en cours, impose à l'opérateur de mettre à jour constamment son image mentale de la situation tactique. Cette reconstruction continue du modèle mental de la situation tactique (Amalberti, 1996) est abordée dans la littérature à travers l'étude du concept de « conscience de la situation », en anglais *Situation Awareness*. Ce chapitre présente un état de l'art de la littérature sur ce concept de *Situation Awareness* afin de comprendre la façon dont est construite la représentation du contexte par l'opérateur et comment évolue cette représentation au fil des changements de tâche.

Dans un premier temps, nous présentons les différentes approches adoptées, parfois en opposition, afin de définir ce concept de *Situation Awareness*, puis nous présenterons les travaux de modélisation intégrant ce concept du point de vue de l'individu mais aussi de l'équipe (dans les systèmes collaboratifs). Enfin, nous présenterons quelques méthodes proposées dans la littérature afin de mesurer le niveau de conscience de situation d'un opérateur.

Note sur la terminologie

Le dictionnaire anglais Oxford définit le terme « *awareness* » (indépendamment du terme « *situation* ») comme la « connaissance ou la perception d'une situation ou d'un fait » et le traduit par le terme français « conscience ». Nous pensons que l'utilisation de « conscience », terme polysémique en français (et à un fort ancrage philosophique), dans la traduction « conscience de la situation » ne traduit pas la neutralité du terme anglais « *awareness* ». La traduction inverse de « conscience » est d'ailleurs donnée par le terme « *consciousness* ». Nous pourrions choisir d'utiliser la traduction « connaissance », que l'on lit dans la définition du Oxford English mais, une fois encore, ce terme ne reflète pas à notre sens l'ensemble des processus sous-jacents à la construction de cette « conscience de la situation ». Nous préférons donc utiliser dans cette thèse le terme anglo-saxon « *Situation Awareness* » (SA).

3.1 LA SITUATION AWARENESS, UN CONCEPT AUX DÉFINITIONS MULTIPLES

Le concept de conscience de la situation (*Situation Awareness*) existe depuis le début du XX^{ème} siècle, où, pendant la Première Guerre Mondiale, il était considéré comme crucial pour les équipages de l'aviation militaire (Press (1986) cité par Endsley (1995b)). De nombreuses définitions de ce concept ont été proposées dans des domaines d'étude comme l'aviation civile ou militaire, le contrôle de trafic aérien, la conduite automobile, ou encore la gestion de centrale nucléaire.

Dans un rapport préparé pour le laboratoire de recherche de l'US Air Force, Dominguez (1994) répertorie 15 définitions différentes du terme *Situation Awareness* et, à partir d'une comparaison des concepts en jeu, elle propose de synthétiser la définition de la *Situation Awareness* comme : « l'extraction continue des informations de l'environnement, leur intégration aux connaissances existantes afin de former un modèle mental cohérent, et l'utilisation de cette image [mentale] pour diriger la perception et anticiper les événements futurs »¹. Ainsi, Dominguez (1994) met en avant les quatre étapes nécessaires à la définition du concept de *Situation Awareness* :

- « extraire les connaissances dans l'environnement »,
- « intégrer ces informations aux connaissances internes existantes afin de créer une image mentale de la situation courante »,
- « utiliser cette image afin de diriger la perception » et
- « anticiper les événements futurs ».

Dans la littérature, la définition de la *Situation Awareness* la plus couramment citée est celle de Endsley (1995b) qui la décrit comme « la perception des éléments de l'environnement dans un volume spatial et temporel, la compréhension de leurs significations, et la projection de leurs états dans le futur proche »². Cette définition linéaire considère la *Situation Awareness* comme le produit des trois niveaux d'analyse de l'information : *perception*, *compréhension* et *projection*. Selon Endsley (1995b), la *Situation Awareness* est un « état de connaissance » qui doit être distingué des processus nécessaires à son acquisition, qu'elle regroupe sous le terme de « *Situation Assessment* » (évaluation de la situation).

Cette dualité entre « état » de la situation et « processus » d'acquisition de la situation constitue un des points les plus discutés de la littérature sur la *Situation Awareness*. Breton et Rousseau (2003) proposent une analyse de 26 définitions du terme *Situation Awareness*, identifiant, pour chacune d'entre elles, leur appartenance

1. traduit de « *Continuous extraction of environmental information, integration of this information with previous knowledge to form a coherent mental picture, and the use of that picture in directing further perception and anticipating future events* »

2. traduit de « *the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future* »

à la catégorie des définitions orientées « état » ou « processus ». La répartition des définitions selon ces deux catégories est équilibrée : 14 définitions sont orientées vers le processus d'acquisition de la situation et 12 voient ce concept comme le produit de cette analyse.

Rousseau *et al.* (2004), proposent d'étudier le concept de *Situation Awareness* selon deux nouvelles approches, associées aux approches *état* et *processus* :

- **l'approche centrée sur l'opérateur** « serait concernée par les propriétés et mécanismes mis en œuvre par l'opérateur lorsqu'il évalue la situation » et correspond aux définitions identifiant la *Situation Awareness* comme un *processus*. Ainsi, ces définitions s'intéressent à « l'ensemble des processus cognitifs mis en jeu dans la production d'une représentation mentale de l'état actuel de la situation ».
- **l'approche centrée sur l'environnement**, « déterminée par l'environnement ou la situation dans laquelle se trouve l'opérateur », correspond aux définitions orientées *état*. La situation peut alors être définie en termes « d'événements, d'objets, de systèmes, des autres personnes et de leurs interactions mutuelles ».

Pew (2000) adopte cette deuxième approche, dans laquelle le concept représente un ensemble de connaissances sur la situation (état du système, phase de l'activité, liste des buts courants, projection de l'état du système), qui changent continuellement et de manière contextuelle. Pew définit trois niveaux de *Situation Awareness* :

- la SA « idéale » correspond aux exigences en termes de connaissances sur la situation que les experts ont définis lors de l'étude du système ;
- la SA « atteignable » (*achievable*) correspond au sous-ensemble des connaissances qui peuvent effectivement être acquises par l'opérateur ;
- la SA « actuelle » correspond à ce que l'opérateur connaît vraiment de la situation.

La conception de nouveaux systèmes doit prendre en compte ces différences et faire en sorte que les informations disponibles à l'opérateur à un instant donné (SA atteignable) correspondent à celles dont il a besoin (SA idéale).

Enfin, pour certains auteurs, la *Situation Awareness* est vue comme une combinaison des deux approches. Par exemple, Smith et Hancock (1995) la décrivent comme « une conscience adaptative et dirigée vers l'extérieur »³. Dans cette définition, la conscience est définie par les auteurs comme « la partie du comportement d'un agent basé sur la connaissance correspondant à la manipulation intentionnelle ». L'approche de Smith et Hancock (1995) est celle d'un couplage entre l'opérateur et son environnement : l'opérateur est guidé par un but inscrit dans son environnement. Ses actions sont guidées par les contraintes de l'environnement et la *Situation Awareness* est vue comme l'invariant du système {opérateur-environnement} géné-

3. « an adaptative and externally directed consciousness »

rant la connaissance et le comportement nécessaires à la réalisation des buts (ces buts se traduisant par un niveau pré-spécifié de performance dans l'environnement).

Cette idée correspond à celles du courant « écologique », rappelant la notion d'affordance des systèmes introduite par (Gibson, 1979), et utilisée dans certains modèles de SA (modèle d'objectif étendu de Kirlik et Strauss (2003), présenté dans le paragraphe 3.2.2.3).

Un état de l'art plus récent de Salmon *et al.* (2008a) compare neuf théories de la *Situation Awareness* parmi les plus citées de la littérature et indique que cette double approche processus / résultat est adoptée quasi-systématiquement (pour 8 théories sur 9).

La multitude des définitions existantes, et dont nous n'avons présenté qu'un extrait, ne remet pas en cause le concept de *Situation Awareness* en soi. Mais cette diversité de définitions a donné lieu à la création de différents modèles décrivant les états et/ou les processus (selon les cas) mis en place dans la détermination des connaissances nécessaires aux opérateurs pour exécuter de manière optimale leur mission.

3.2 LES MODÈLES DE SITUATION AWARENESS

Nous proposons dans cette section de présenter une sélection de modèles de *Situation Awareness* (SA), catégorisés dans deux groupes : les modèles descriptifs, adoptant une approche empirique de la SA, et les modèles prescriptifs, correspondant à une approche plus théorique et « informatique » (Rousseau *et al.*, 2004).

3.2.1 Modèles descriptifs de SA

D'après la définition de Bryant *et al.* (2004), les modèles descriptifs de *Situation Awareness* « tentent de fournir une large vue d'ensemble des différentes étapes de traitement de l'information lorsqu'une décision est prise ou une action engagée ». Ces modèles décrivent la SA de manière empirique, afin d'approcher au mieux les processus mis en jeu. Les modèles descriptifs représentent la majorité des modèles de la littérature, et proposent une « description globale et systématique de la *Situation Awareness* et de la façon dont elle est construite » (Rousseau *et al.*, 2004). Nous décrivons deux de ces modèles : le modèle le plus connu, proposé par Endsley, qui adopte une approche par le traitement de l'information et le modèle RPD (Klein, 1993) issu de la théorie de la « décision naturelle » (*Naturalistic Decision Making*).

3.2.1.1 Le modèle à 3 niveaux de Endsley

Endsley (1995b, 2000c) nous propose un modèle basé sur sa définition de la *Situation Awareness* (section 3.1) en intégrant les trois niveaux de cette définition au sein d'un modèle plus large (figure 3.1). Ce modèle est constitué de deux parties :

- Le *Core SA model* (bloc grisé sur la figure 3.1) décrit les 3 niveaux de la *Situation Awareness* (perception, compréhension, projection décrit ci-dessous) et le processus de prise de décision / action attaché à la SA ;
- Les différents facteurs impactant la *Situation Awareness* sont représentés autour du *Core model*. Ces facteurs sont issus du système avec lequel interagit l'opérateur (capacités, nature des interfaces...), des tâches réalisées (complexité, niveau d'automatisation...) mais peuvent également être des facteurs individuels (expérience, entraînement...) ou liés à la mission (buts et objectifs, attentes suite au briefing par exemple).

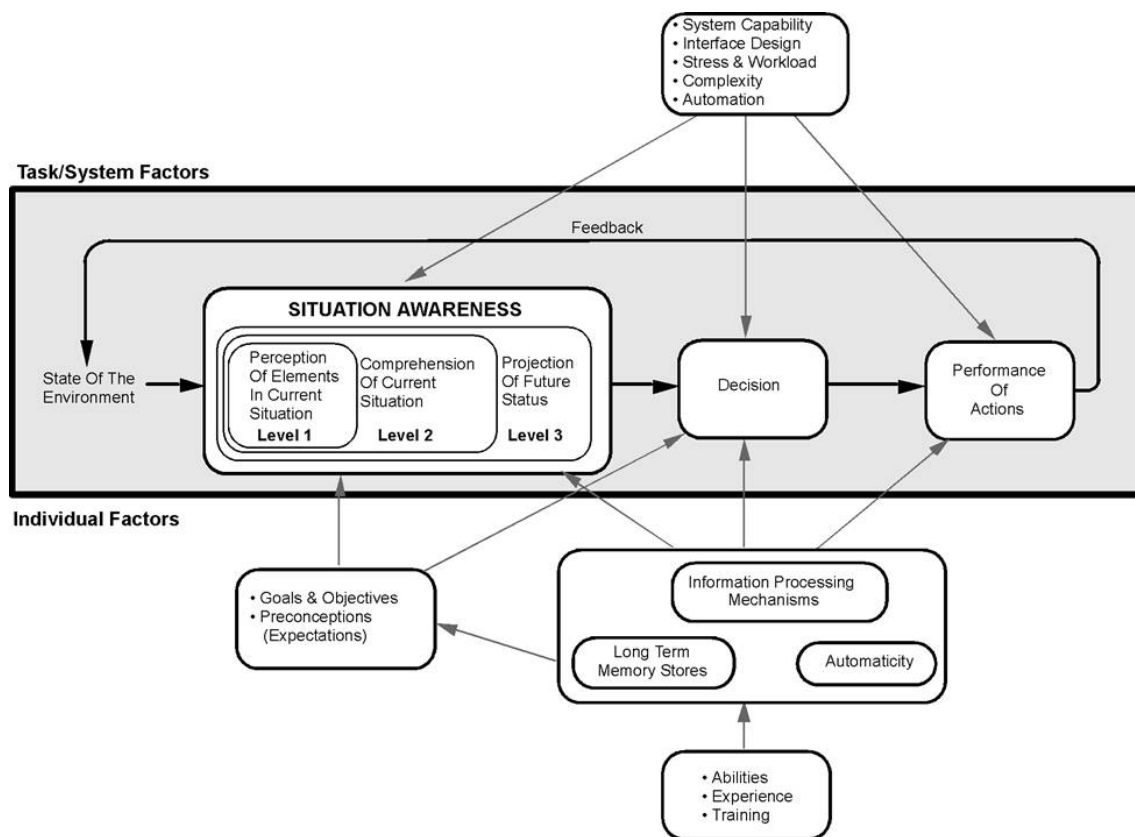


FIGURE 3.1 : modèle à 3 niveaux de Endsley (1995b) (adaptation).

Le **niveau 1** du modèle de Endsley, première étape de construction de la SA, correspond à la **perception des éléments de la situation courante**. Il fournit des « informations sur les états, les attributs et la dynamique des éléments pertinents de l'environnement » (Endsley, 1995b). Aucune analyse des informations n'est effectuée, ce niveau gère notamment le focus attentionnel de l'opérateur qui peut être dirigé vers la capture de certains éléments associés aux objectifs par exemple (aspect « *top-down* » ou « endogène ») ou attiré par les éléments saillants pertinents (aspect « *bottom-up* » ou « exogène »).

Ensuite, l'opérateur aborde le **niveau 2**, la **compréhension de la situation courante**, qui « synthétise les éléments disjoints du premier niveau » afin de fournir une image organisée de cette situation, à travers la compréhension de la signification des objets et événements perçus. Ce deuxième niveau constitue un modèle « situationnel » décrivant l'état actuel du modèle mental que se fait l'opérateur de la situation.

Enfin, le **niveau 3**, la **projection des états futurs**, propose un pronostic de l'état futur du système en s'appuyant sur les éléments perçus (niveau 1), injectés dans les modèles mentaux du niveau 2, afin d'obtenir des situations probables et permettre à l'opérateur d'anticiper sur cet état futur.

Endsley (1995b) souligne l'importance des modèles mentaux aux 3 niveaux de son modèle : ils permettent de diriger l'attention de l'opérateur au niveau 1, d'intégrer les éléments afin de fournir une image compréhensible de la situation courante (niveau 2) et probable (niveau 3).

Le modèle de Endsley *et al.* (2000a) le plus complet de la littérature est présenté figure 3.2, il s'agit d'un modèle de *Situation Awareness* appliqué aux fantassins de l'armée américaine. On y retrouve les quatre groupes de facteurs impactant la SA : le monde extérieur (*external world*), les facteurs environnementaux et liés aux tâches (*Task & environmental factors*), les facteurs individuels (*individual factors*) et les facteurs du domaine (relatif à la mission, aux amis / ennemis...).

Différentes versions du modèle linéaire à 3 niveaux ont été opérationnalisées à plusieurs reprises. Endsley propose notamment une méthodologie d'extraction de connaissances, et des exigences en matière de SA permettant d'appuyer la construction de ce modèle. Cette méthodologie consiste à définir les buts et sous-buts de l'activité étudiée (analyse appelée par Endsley (1993) « *Goal Directed Task Analysis* », GDTA), puis de définir les exigences pour chacun des 3 niveaux du modèle de SA. Elle a été appliquée, entre autres, à l'analyse de l'activité de pilotes de ligne (Endsley *et al.*, 1998a), de pilotes militaires en situation de combat aérien (Endsley, 1993) et de contrôleurs aériens (Endsley et Rodgers, 1994).

3.2.1.2 Le modèle RPD (*Recognition Primed Decision*)

Le courant *Naturalistic Decision Making* (NDM) définit le processus décisionnel comme « la façon dont les gens utilisent leur expérience pour prendre des décisions dans un contexte réaliste » (Zsombok, 1997) c'est-à-dire dans la réalité de l'exercice de leur activité. Pour Klein (1993), qui se revendique de ce courant, la *Situation Awareness* est un « processus de recherche d'information actif et guidé » plutôt qu'un simple stockage de détails sur la situation (définition orientée processus). Cette idée est mise en œuvre dans le modèle RPD (*Recognition Primed Decision*) qui décrit 3 modes d'évaluation (figure 3.3) de la situation. Le principe de ce modèle est de permettre la prise de décisions par l'opérateur sans comparaison des différentes options possibles (chaque situation entraîne un cours d'action) :

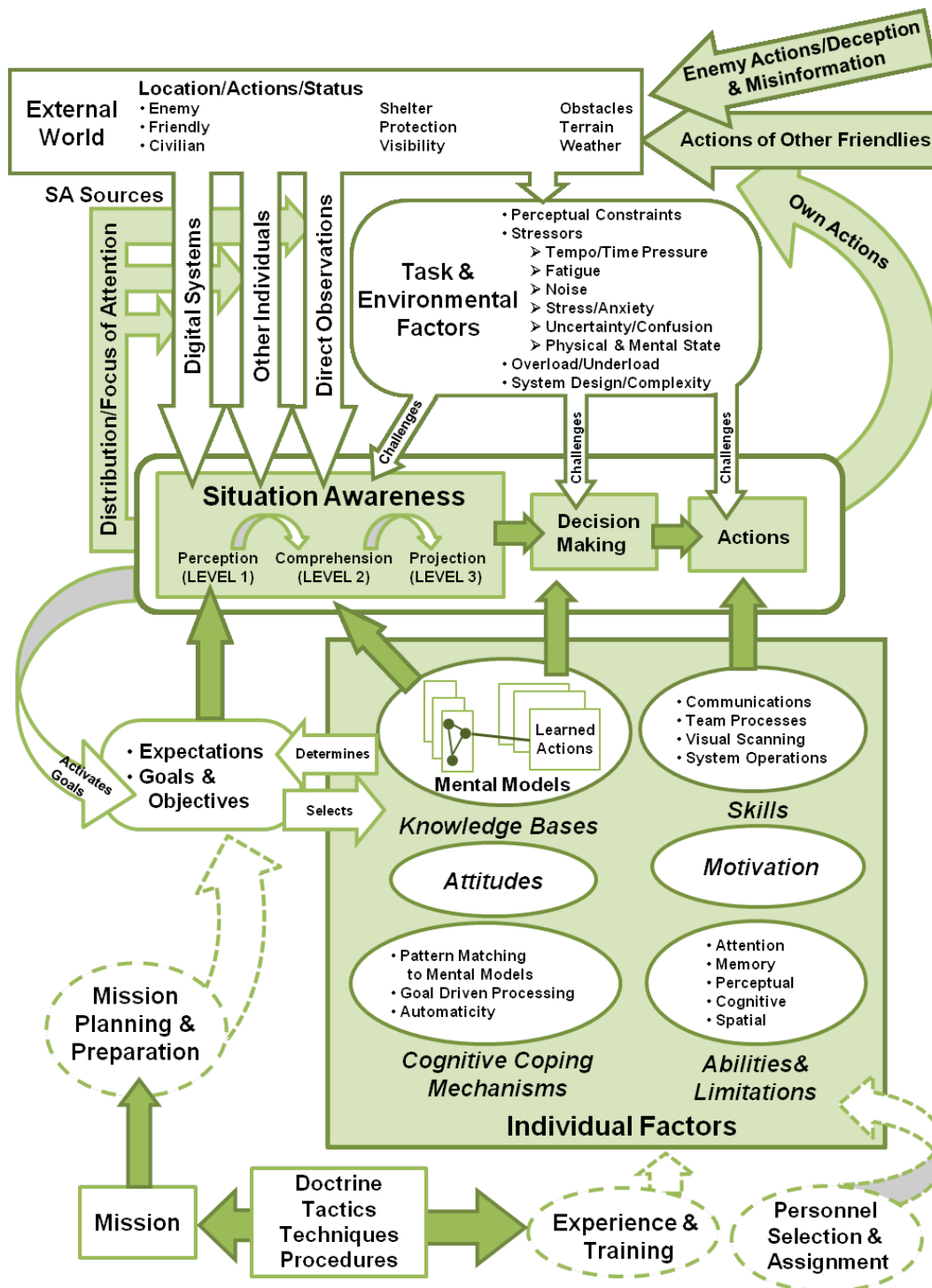


FIGURE 3.2 : modèle de Endsley *et al.* (2000a) appliqué au fantassin (adaptation).

Reconnaissance simple (figure 3.3a) : l'expert reconnaît la situation et enclenche le cours d'action adéquat pour répondre à la problématique identifiée. Aucune autre alternative n'est évaluée.

Développement d'un cours d'action (figure 3.3b) : l'expert reconnaît la situation et effectue un diagnostic interne des différentes étapes du schéma associé habituellement à cette situation. Si la simulation interne donne un résultat jugé probant, le schéma est enclenché, sinon il peut être adapté.

Stratégie complexe (figure 3.3c) : lorsque la situation n'est pas reconnue ou que le niveau 2 a échoué (les cours d'action connus n'ont pu être adaptés à la situation), deux cas de figure se présentent. Dans le premier cas, la situation est ré-évaluée (bloc "*Reassess Situation*") et dans le deuxième cas, un autre cours d'action est sélectionné.

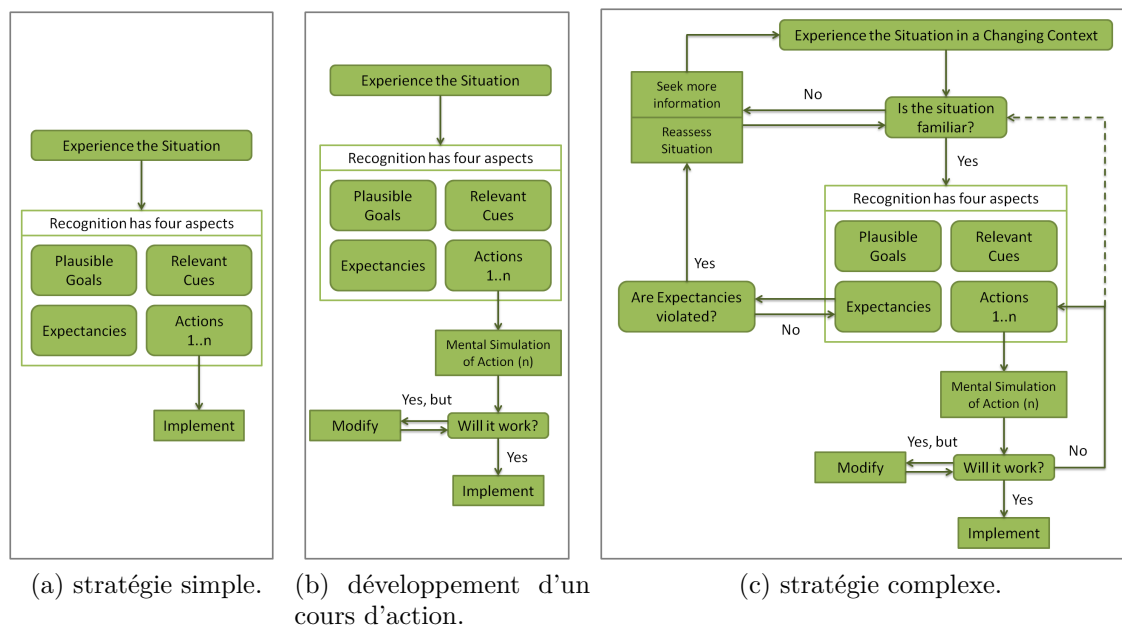


FIGURE 3.3 : les 3 modes d'évaluation du modèle *Recognition Primed Decision* de Klein (1993).

La *Situation Awareness* est « centrale aux trois versions du modèle RPD » (Klein, 2000), elle constitue la clef de la première étape du processus décisionnel décrit dans ce modèle. En effet, quelque soit la stratégie adoptée par l'opérateur (parmi les 3 identifiées), le cours d'action est activé, après avoir éventuellement été adapté, après la phase de reconnaissance de la situation. Les quatre aspects de la reconnaissance de situation peuvent se projeter sur deux des trois niveaux du modèle de Endsley (1995b) :

- les buts plausibles, les indications pertinentes et les actions « raisonnables » associées à cette situation correspondent tous les trois au niveau 2 du modèle Endsley, la compréhension de la situation ;
- le dernier aspect, les attentes (*expectancies*) générées lorsque la situation a été reconnue comme « typique » correspond au niveau 3 du modèle de Endsley, la projection dans le futur.

Ainsi, selon Klein (2000), le modèle de Endsley et le modèle RPD sont compatibles, même si ce dernier n'intègre pas directement le niveau 1 dans sa représentation, il prend en compte la *Situation Awareness* de l'opérateur dans le processus décisionnel.

3.2.2 Modèles prescriptifs de SA

A la différence des modèles descriptifs, les modèles prescriptifs adoptent une approche plus formelle de la représentation des processus et de leurs résultats. Selon Bryant *et al.* (2004), les modèles prescriptifs « identifient les règles et les heuristiques qui guident les opérations au sein et entre les différentes étapes du traitement de la situation ». Cette approche de modélisation peut être considérée comme plus « concrète » car elle intègre la représentation des éléments de situation dans un modèle formalisé, induisant la construction de structures de données. Nous présentons dans la suite de cette section trois modèles prescriptifs de SA.

3.2.2.1 L'outil MIDAS et le modèle de la performance humaine

La NASA Smith et Tyler (1997); Tyler *et al.* (1998) a développé un outil, appelé **MIDAS** (*Man-Machine Design and Analysis System*), qui permet notamment d'étudier les impacts des technologies d'assistance à l'opérateur en utilisant des modèles de performance humaine (*Human Performance Model*, HPM) pour simuler l'effet de différentes solutions sur la performance de l'opérateur d'un système complexe. MIDAS a été appliqué à un panel très large d'activités de la NASA depuis ses débuts : aviation civile et militaire, navette spatiale et station spatiale internationale⁴. MIDAS est conçu sur la base de deux modèles principaux, le modèle du domaine et le modèle de l'opérateur ((Smith et Tyler, 1997; Tyler *et al.*, 1998) pour la version originale de l'architecture, (Gore, 2010) pour une version plus récente) :

- Le **modèle du domaine** décrit l'ensemble des éléments gravitant autour de la salle de commande du système étudié et leurs liens avec celui-ci, à savoir l'environnement extérieur, les véhicules mais également l'équipage et leur mission ;
- Les membres de l'équipage sont modélisés par un **modèle de l'opérateur** (figure 3.4) intégrant la simulation des entrées (perception) / sorties (actions motrices), une gestion de l'attention et des mémoires (mémoire de travail et

4. voir aussi <http://humansystems.arc.nasa.gov/groups/midas/>

mémoire à long terme).

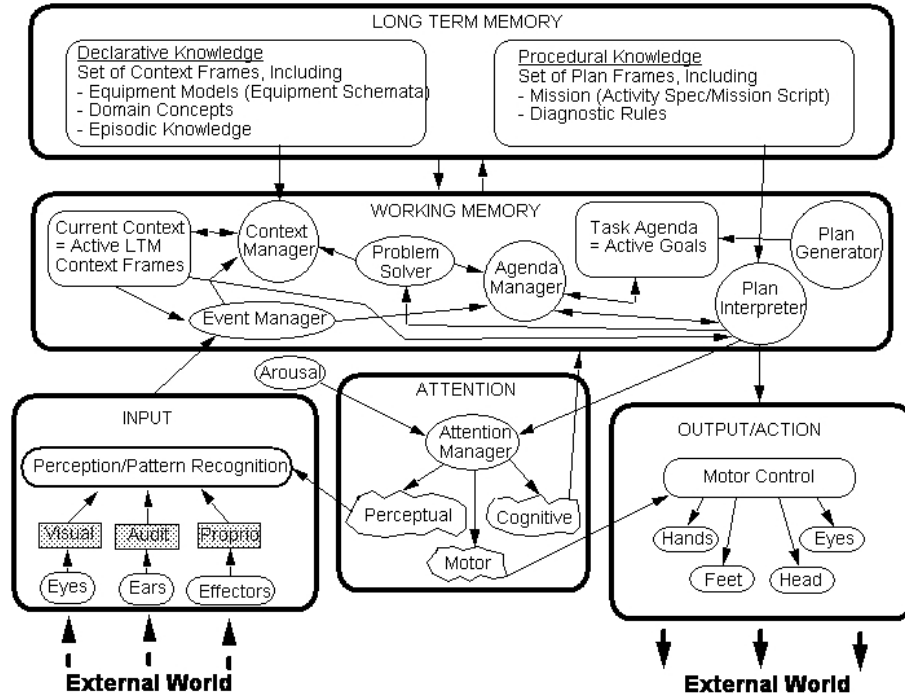


FIGURE 3.4 : modèle opérateur du logiciel MIDAS (extrait de Smith et Tyler (1997)).

Shively *et al.* (1997) ont intégré au logiciel MIDAS une brique permettant de calculer la *Situation Awareness* de l'opérateur. Les éléments de situation (*Situational Elements*, SE), c'est-à-dire les éléments qui composent l'environnement définissant la situation sont pondérés en fonction de leur pertinence pour la situation (une vache dans un champ aura moins d'importance qu'un soldat ennemi dans un contexte de combat terrestre par exemple). 4 niveaux de SA sont ensuite définis pour chaque élément : détection, reconnaissance, identification et compréhension. La combinaison de la pertinence et de la SA associée à un élément permet de lui attribuer un score.

MIDAS calcule ensuite différents scores de SA, ramenés entre 0 et 1 : un score de SA « perçue » (la SA de l'opérateur), une SA « actuelle » (prenant en compte les éléments importants non perçus) et une SA « d'erreur » (score des éléments sur lequel l'opérateur s'est trompé : mauvaise perception, ou interprétation par exemple). Le score idéal est défini à la valeur 1. Ainsi, on retrouve la définition de Pew (2000), le but est d'approcher au plus près la SA « actuelle » de la SA « idéale ».

L'intégration du modèle de SA dans MIDAS n'est pour l'instant liée qu'au premier niveau du modèle de Endsley (perception), celle-ci a été améliorée avec l'ajout du modèle d'allocation de l'attention SEEV (*Salience, Expectancy, Effort, Value*) de Wickens *et al.* (2003) qui permet de calculer, pour chaque élément, la probabilité

pour l'opérateur d'accorder son attention à cet élément selon la formule

$$P(\text{Attention}) = a \times \text{Saliency} - b \times \text{Effort} + c \times \text{Expectancy} + d \times \text{Value}$$

où la **saillance** (*saliency*) « capture le contraste de l'élément dans la scène », l'**effort** capture la charge de travail courante de l'opérateur (imposée par d'autres tâches concurrentes) ainsi que l'effort requis pour déplacer l'attention visuelle, l'**attente** (*expectancy*) traduit la pertinence d'occurrence de l'élément, plus le pilote s'attend à voir l'élément, plus il va chercher à le voir, et la **valeur** (*value*) correspond à l'importance donnée à l'élément pour la tâche courante par l'opérateur. À partir de ce nouveau modèle, des travaux d'analyse prédictive et comparative d'intégration d'interfaces dans les cockpits ont été initiés au sein de MIDAS (Hooey *et al.*, 2011).

3.2.2.2 Le modèle SAMPLE

Le modèle **SAMPLE** (*Situation Awareness Model for Pilot-in-the-Loop Evaluation*) proposé par Zacharias *et al.* (1996) est focalisé sur la modélisation de la SA, élément central du modèle du pilote. Soutenu par un logiciel de simulation à destination des concepteurs de systèmes dynamiques, ce modèle est composé de quatre composants (figure 3.5) : le modèle du pilote, le modèle de l'avion, le « générateur de métriques de SA et de performance » et l'interface utilisateur (GUI).

Basé sur des modèles mentaux construits sous forme de réseaux bayésien de croyances, le **modèle du pilote** est centré sur la SA. Il intègre les phases d'analyse de l'information, d'évaluation de la situation et de prise de décision du pilote à partir des informations reçues par les sous-systèmes. Ces réseaux de croyances sont constitués d'éléments de situation et d'événements, qui permettent, à partir de l'évaluation de la situation courante de calculer des probabilités de situations futures et d'ainsi choisir la meilleure marche à suivre.

Le modèle de pilote reçoit ses informations des modèles des **avions et autres sous-systèmes**, qui simulent le comportement de ces derniers afin de fournir des informations opérationnellement pertinentes sur les événements et la situation courante.

Le **générateur de métriques SA et de performance** permet d'évaluer les performances et la SA du pilote en comparaison avec un modèle de référence, générant des performances et une SA idéales. Pour les 3 phases du modèle pilote (analyse de l'information, évaluation de la situation et prise de décision), un score de disparité est déterminé, en calculant la différence normalisée entre les valeurs produites par le modèle du pilote recevant une information parfaite (valeurs de référence) et le modèle du pilote qui reçoit les informations « réelles » du système. Le score de disparité permet, pour chaque sous-système évalué, de déterminer la performance et la SA « actuelle » du pilote.

Enfin, l'ensemble de ces composants est paramétrable à travers l'**interface utilisateur (GUI)**, permettant au concepteur de contrôler la simulation.

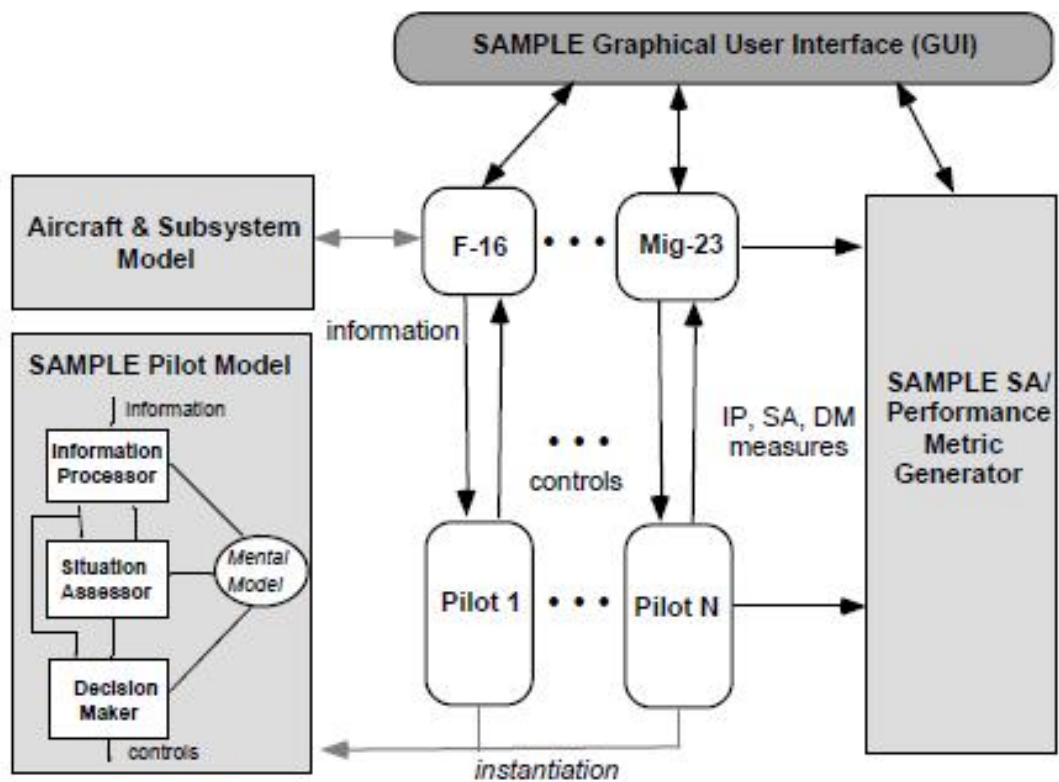


FIGURE 3.5 : modèle SAMPLE de Zacharias *et al.* (1996).

A la différence du modèle intégré à MIDAS, le modèle SAMPLE ne prend pas en compte l'aspect « perception » de la *Situation Awareness*. Nous remarquons également que ce modèle est très spécifique au système étudié. Bien que son principe pourrait être adapté à d'autres systèmes, il a été développé spécifiquement pour l'évaluation de solutions à destination des pilotes.

3.2.2.3 Une approche « système » de la SA

Kirlik et Strauss (2003) présentent une approche de modélisation mettant en application l'aspect dual (état / processus) de la *Situation Awareness*. En effet, les auteurs basent leur modèle sur la théorie du jugement humain incertain (Brunswik, 1956), cette incertitude découlant de l'utilisation d'une interface (ici technologique) entre l'environnement et l'opérateur, notamment à travers les indications (*cues*) affichées et reflétant l'état de l'environnement. Dans cette approche orientée système de la SA, la construction de la SA par l'interaction est favorisée par l'aspect écologique (au sens de Gibson (1979)) de l'interface.

Le modèle proposé est calqué sur le modèle initial de Brunswik (1956), présenté figure 3.6 : la qualité du jugement de l'opérateur sur la situation à travers une interface est mesurée par le calcul d'un coefficient de corrélation entre la réalité (côté environnement, à gauche du schéma) et la situation jugée (côté opérateur, à droite du schéma). La valeur de ce coefficient est modifiée par l'incertitude induite par l'interface.

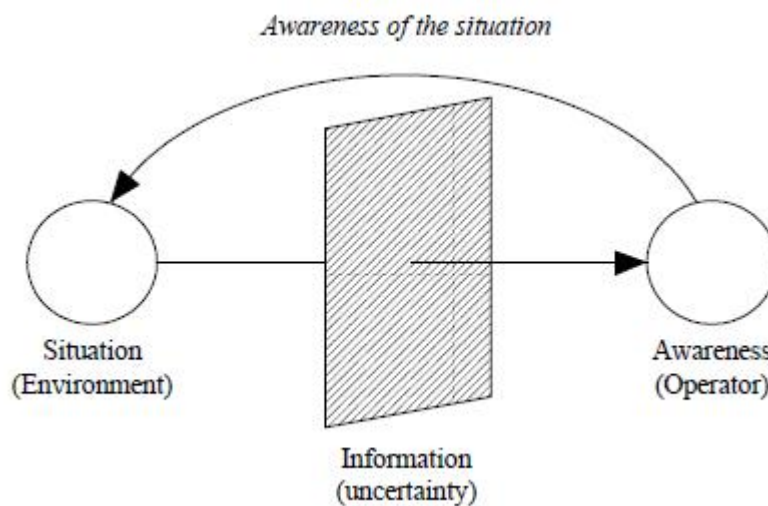


FIGURE 3.6 : la représentation du jugement selon Brunswik (1956)

Plusieurs modèles se sont succédés afin d'améliorer le calcul linéaire de corrélation de Brunswik (1956), en précisant notamment la décomposition de la chaîne d'extraction et de transmission de la situation. Nous présentons la dernière version

de ce modèle, le « modèle d'objectif étendu » (*Expanded Lens Model* ou ELM) (Stewart et Lusk, 1994). Ce modèle (figure 3.7) décompose la chaîne de traitement de l'information en plusieurs étapes : capture par le système, affichage et représentation par l'opérateur.

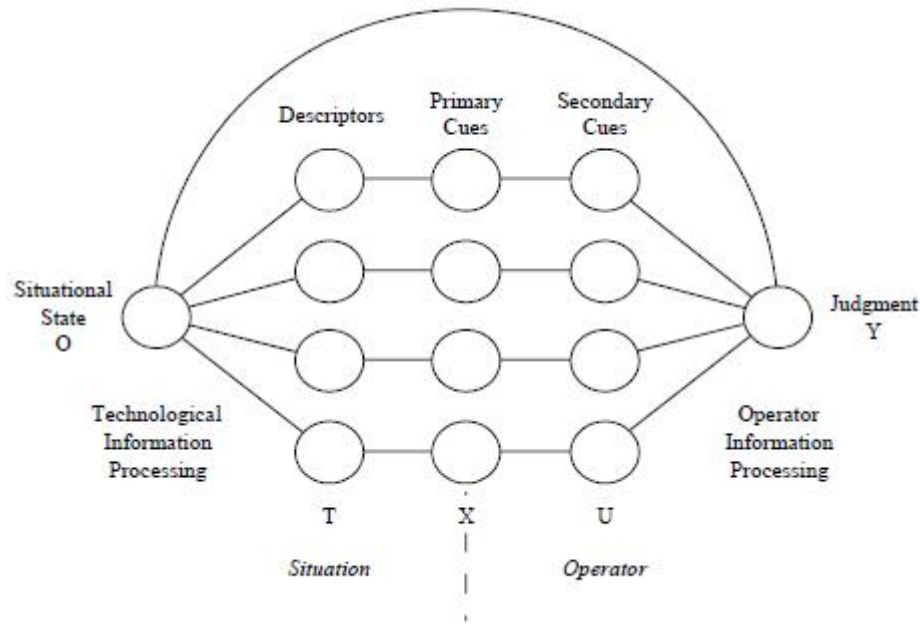


FIGURE 3.7 : *Expanded Lens Model* de Stewart et Lusk (1994) (adapté de Kirlik et Strauss (2003))

Un score d'aptitude global (*skill score*, SS) est calculé par l'équation 3.1 :

$$SS = [(R_{OT})(V_{TX})(G)(V_{UX})(R_{YU})]^2 - \left[r_{YO} - \left(\frac{s_Y}{s_O} \right) \right]^2 - \left[\frac{(\bar{Y} - \bar{O})}{s_O} \right]^2 \quad (3.1)$$

Ainsi, à chaque segment de la figure 3.7 correspond un coefficient (première partie de l'équation 3.1) :

- Les deux premiers segments du modèle (O-T-X) traduisent la transformation de l'information par le système :
 - La première étape (segment OT) correspond à l'extraction des informations par le système dans l'environnement, le coefficient R_{OT} caractérise la corrélation des données décrites avec la réalité. Par exemple, dans un environnement bruyé, les capteurs peuvent avoir une mauvaise « image » de la réalité, influençant l'ensemble de la chaîne de traitement.
 - Ces informations sont ensuite fournies à l'opérateur (segment TX) par les indications (*cues*) de l'interface du système, le coefficient V_{TX} associé à ce segment traduit la fidélité de représentation du système par rapport aux informations.

- Symétriquement aux segments OT et TX, on retrouve les mêmes processus implémentés du côté « opérateur » :
 - Le segment XU correspond à l'assimilation de l'information par l'opérateur. Celui-ci s'apparente au niveau 1 du modèle de Endsley (1995b) et est caractérisé par un coefficient V_{UX} qui traduit la cohérence entre les indications du système (*primary cues*) et celles que l'opérateur capture (*secondary cues*). Un coefficient faible traduit une mauvaise perception de l'indication par l'opérateur.
 - Le dernier segment, UY, correspond au jugement de l'opérateur sur les informations collectées, R_{YU} traduit la cohérence du jugement de situation par rapport aux indications perçues. Ce segment correspond au niveau 3 du modèle de Endsley.
- Enfin, le facteur G représente l'adaptation de l'opérateur à la situation : il est appelé facteur de « connaissance » dans le *Lens model* et correspond à la corrélation initiale entre le modèle de situation et le modèle de l'opérateur.
- Les deux composantes $\left[r_{YO} - \left(\frac{s_Y}{s_O}\right)\right]^2$ et $\left[\frac{(\bar{Y}-\bar{O})}{s_O}\right]^2$ de l'équation 3.1 correspondent à deux mesures de biais fondées sur les caractéristiques de l'opérateur (à sous-estimer ou sur-estimer les situations notamment).

Ainsi, les 7 composantes du modèle permettent de mesurer la SA pour chaque élément pertinent de la tâche en cours (défini comme un critère d'acquisition lors de la modélisation). Le travail de modélisation associé à ce modèle est fastidieux car il est nécessaire, pour toutes les composantes, de définir une valeur quantitative, ce qui peut être difficile si l'élément correspondant n'a pas de mesure quantifiable associée.

Dans un deuxième article, Strauss et Kirlik (2003) présentent une expérimentation appliquant leur modèle à une tâche opérationnelle au cours de laquelle, sur une période de 8 jours, les sujets doivent exprimer leur jugement sur les paramètres de croisière d'un sous-marin dont ils sont des membres d'équipage. La comparaison porte sur l'analyse de deux interfaces : une interface « basique » et une interface « augmentée » et la variation de certains coefficients de l'équation 3.1 pour chacune des interfaces. Les résultats permettent de mettre en exergue la différence de *Situation Awareness* entre les sujets selon leurs performances respectives et le type d'interface utilisée.

3.2.3 Synthèse sur les modèles de SA individuelle

A travers ces deux premières sections, nous avons pu observer que la littérature sur le concept de *Situation Awareness* propose un nombre impressionnant de définitions différentes pour le concept de *Situation Awareness*, englobant chacune un ensemble de processus et concepts de la psychologie cognitive et du domaine des facteurs humains.

Pour résumer, Moray (2004, page 4) écrit que la « *Situation Awareness* n'est pas

vraiment le nom d'une fonction psychologique particulière. C'est un *raccourci* pour décrire l'action d'être à l'écoute de ce qui se passe autour de soi dans un environnement complexe et dynamique. Une telle tâche se fait en utilisant toutes ses capacités cognitives ».

En termes de modèles, la littérature est principalement focalisée sur les modèles descriptifs du concept de *Situation Awareness* qui décrivent les différentes fonctions cognitives en jeu, une vue d'ensemble du processus de construction de la SA, au détriment des modèles prescriptifs, plus proches de l'aspect informatique et ingénierie. Le faible nombre de modèles prescriptifs est souligné par Moray (2004) qui préconise un rapprochement des domaines de l'ingénierie et des facteurs humains afin de proposer de tels modèles.

Ces modèles s'intéressent à la SA individuelle d'un point de vue global, et ne prennent pas en compte les multiples tâches des missions de l'opérateur, qui nécessitent chacune un modèle spécifique de la SA afin d'atteindre les différents buts associés à ces tâches. Dans la suite de ce chapitre, nous nous intéressons ainsi à la façon dont le concept de *Situation Awareness* a été adapté à l'étude des systèmes collaboratifs, où chaque opérateur adopte un point de vue différent sur la situation tactique en fonction de sa tâche.

Enfin, plusieurs méthodes de mesures de la SA ont été proposées, nous les présentons dans la section 3.4.

3.3 LA SITUATION AWARENESS DANS LES SYSTÈMES COLLABORATIFS

L'étude du concept de *Situation Awareness* dans les systèmes collaboratifs introduit la notion de travail en équipe et introduit un niveau de complexité supérieur à l'étude d'un concept parfois déjà mal compris au niveau individuel. Salas *et al.* (1992) (repris de Salas *et al.*, 1995) définit une équipe comme « un groupe de deux personnes distinctes ou plus, interagissant dynamiquement de manière interdépendante et adaptative pour atteindre un objet, un but, une mission communs, et ayant chacune un rôle ou des fonctions spécifiques, pour un laps de temps donné ».

L'analyse de la *Situation Awareness* dans les systèmes collaboratifs considère ainsi plusieurs points de vue sur une même situation, chaque membre de l'équipe devant connaître les informations nécessaires à la réalisation de sa tâche. Dans le cadre de notre étude, cette approche à points de vue multiples nous semble intéressante car elle s'apparente aux multiples points de vue que peut adopter un opérateur unique sur les différentes tâches entre lesquelles il doit alterner.

Dans cette section, nous présentons deux approches de définition des concepts de SA dans les systèmes collaboratifs : le concept de SA en équipe (*Team SA*) et celui de SA distribuée (*Distributed SA*).

3.3.1 La SA en équipe

Comme le notent Salas *et al.* (1995), l'étude de la SA d'une équipe (*Team Situation Awareness* ou TSA) ne se résume pas à la simple étude des SA individuelles (telles que définie dans les sections 3.1 et 3.2) et à leur combinaison mais introduit des dimensions supplémentaires, comme la coordination et le partage d'informations entre les membres d'une équipe. Il s'agit pour chaque opérateur de l'équipe de construire leur propre SA, une SA individuelle, et de mettre en place des processus de communication et de collaboration afin de partager leur SA avec les autres membres de l'équipe. Ainsi, le concept de TSA est présenté comme multidimensionnel (Endsley, 1995b; Salas *et al.*, 1995; Nofi, 2000) et illustré par des diagrammes de Venn (figure 3.8) :

- La **Situation Awareness individuelle** (correspondant aux définitions de la section 3.1), représentée par les 3 ensembles A, B et C ;
- La **SA partagée** (*Shared SA* ou SSA), qui correspond à l'intersection des différents ensembles. Cette SA peut être partagée par quelques opérateurs (AB, AC, BC) ou par l'ensemble des membres d'une équipe (ABC). Endsley et Jones (1997) la définissent comme « le degré auquel les membres de l'équipe ont le même niveau de SA pour des exigences partagées de SA ». Dans le contexte de la TSA, cette dimension sous-entend que certaines exigences de SA et buts associés sont partagés entre les membres de l'équipe.
- La **SA globale de l'équipe** entière, la *Team SA*, qui correspond à l'union des SA individuelles de chaque membre et représente l'ensemble des éléments de situation intégrés par ces opérateurs. Elle est définie par Endsley (1995b) comme « le degré auquel chaque membre de l'équipe possède une SA nécessaire à l'accomplissement de sa tâche ».

La réussite d'un travail en équipe s'appuie sur le fait que chaque membre possède une SA individuelle suffisante eu égard aux exigences définies pour sa tâche mais également que les différents membres aient la même SA sur les éléments partagés : une SA partagée différente peut entraîner des incompréhension et des erreurs. Cette double exigence est traduite par Saner *et al.* (2009) en introduisant un degré de précision (*accuracy*) de la SA individuelle (par rapport aux exigences définies pour réaliser les tâches de l'opérateur) et un degré de similarité entre la SA de deux individus (nombre d'éléments de situation partagés en fonction des exigences communes).

Dans un état de l'art sur la SA, Salmon *et al.* (2008a) présentent cette idée de décomposition tripartite de la SA en équipe comme une combinaison de la SA individuelle propre à un opérateur, de la SA associée aux autres membres de l'équipe (la *taskwork SA*) et de la SA globale de l'équipe (la *teamwork SA*). La figure 3.9 présente la façon dont ces composantes sont liées au sein d'un modèle de SA en équipe.

Dans la littérature sur la SA en équipe, la construction de la SA individuelle s'appuie généralement sur le modèle linéaire à 3 niveaux (perception, compréhension,

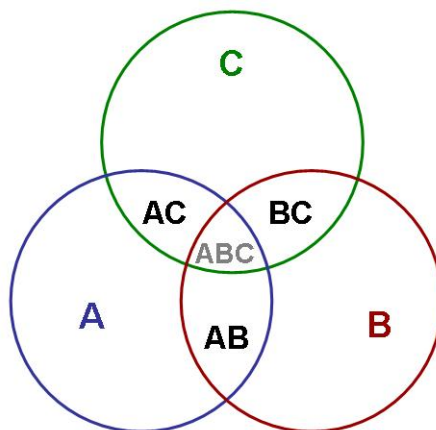


FIGURE 3.8 : schématisation de la SA d'une équipe de 3 membres (adapté de Nofi, 2000).

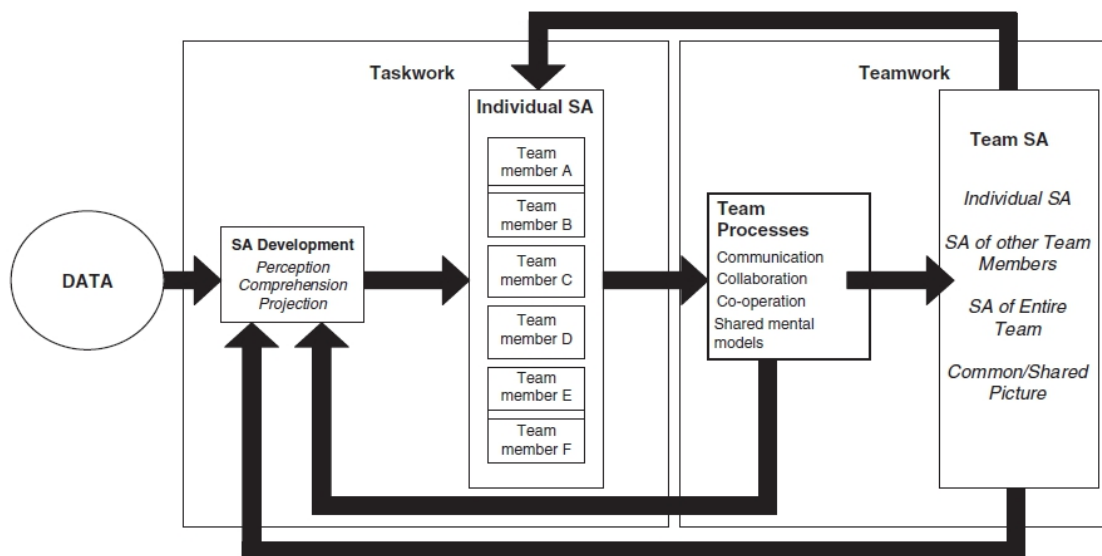


FIGURE 3.9 : modèle de la SA par équipe résumé par Salmon *et al.* (2008a).

projection) de Endsley (1995b). Les processus enclenchés au sein de l'équipe afin de partager la SA et de construire une SA globale (communication, collaboration, coopération) permettent aux membres de l'équipe d'outrepasser les limites de leurs perception et de leur modèles mentaux afin de comprendre au mieux la situation (Salas *et al.*, 1995).

Enfin, les processus de partage de la connaissance ne sont pas les seuls paramètres permettant de construire une « bonne SA ». En effet, Bolstad *et al.* (2005) proposent une étude dans laquelle ils prennent en compte la proximité géographique, la similarité d'expérience (grade, appartenance à un même corps d'armée, expériences de

missions passées) et la fréquence de communication lors des sessions précédentes entre les membres d'une même équipe afin d'étudier l'impact de ces 5 facteurs sur la construction de la SA. Ils indiquent que la proximité géographique (colocalisation) est le facteur majeur de construction d'une bonne SA partagée.

3.3.2 La SA distribuée

3.3.2.1 Définition

La théorie de la SA distribuée (*Distributed Situation Awareness* ou DSA), introduite par Stanton *et al.* (2006), représente une alternative à la théorie de la *Team SA*. Celle-ci se repose sur la théorie de la cognition distribuée (Hutchins, 1995) et s'appuie sur les travaux de Artman et Garbis (1998) décrivant la SA d'une équipe comme une caractéristique du système et non comme une compréhension partagée de la situation. La SA du système collaboratif est ainsi distribuée entre les agents humains et technologiques impliqués, elle est représentée par un réseau d'informations, sur lequel chaque composante du système possède une vue différente.

Selon Stanton *et al.* (2009), « l'approche de la SA partagée (*shared SA*) peut mal orienter l'attention de l'opérateur vers des aspects non pertinents de la tâche ». En effet, celle-ci implique un partage entre les opérateurs des exigences sur les éléments de situation et un partage des buts. A l'inverse, la DSA autorise des exigences et buts différents, mais potentiellement *compatibles* à certains instants de la mission. A ces instants, le partage des connaissances n'est pas exclu du modèle et peut se faire par l'intermédiaire de *transactions* entre les agents du système (communication, interaction). Stanton *et al.* (2006) définissent la DSA ainsi :

“

La DSA est constituée des connaissances activées pour une tâche spécifique dans un système donné.

(Stanton *et al.*, 2006)

”

La *connaissance* (traduit de l'anglais « *knowledge* ») représente ici les connaissances des individus mais également les données du système (par exemple intégrées à travers des capteurs) reflétant l'état de l'environnement et de ses changements. Elle correspond à l'idée de schéma activé (au sens de la théorie des schémas, sur laquelle s'appuient Stanton *et al.* (2009)) pour une tâche donnée et qui sera mis à jour de manière interactive à travers un cycle « perception-action » (Neisser, 1976) permettant d'intégrer les changements du monde et de les répercuter sur la situa-

tion. Appliquées à l'individu, nous retrouvons les idées de la théorie de la SA de Smith et Hancock (1995) (section 3.1).

3.3.2.2 La méthodologie des réseaux propositionnels

La théorie de la DSA a été étudiée dans le domaine de la navigation maritime militaire à travers la modélisation de l'activité d'une passerelle de frégate de la *Navy* britannique (Stanton *et al.*, 2006, 2009), la distribution d'énergie (Salmon *et al.*, 2008b) ou encore la planification de conflit terrestre (Salmon *et al.*, 2010). Stanton *et al.* (2006, 2009) proposent un modèle s'appuyant sur trois représentations.

- un **réseau social** présentant les liens entre les membres de l'équipe et les moyens de communication utilisés ;
- un **réseau de tâches** présentant les tâches (orientées vers les buts à réaliser) des différents agents du système. Plusieurs instances d'une même tâche peuvent être exécutées en parallèle par différents agents du système
- un **réseau de connaissances** présentant l'ensemble des informations, ainsi que leurs relations, que le système (comprenant les agents humains et technologies) doit connaître pour remplir avec succès sa mission.

La construction de ces différents réseaux s'appuie sur l'extraction des connaissances des opérateurs et experts du domaine (*System Matter Experts* ou SME). Cette construction suit une méthodologie dite « des réseaux propositionnels » (terme introduit dans Salmon *et al.*, 2010), divisée en plusieurs étapes que nous présentons dans les paragraphes suivants.

Étape 1 : extraction des connaissances

La phase d'extraction de connaissances s'appuie sur les observations ainsi que des entretiens. Stanton *et al.* (2006) suivent le protocole d'entretien *Critical Decision Method* (Klein, 2000; Crandall *et al.*, 2006) qui consiste à poser une série de questions à l'opérateur au sujet d'un incident non routinier et nécessitant un jugement expert ou une prise de décision afin d'être résolu. Ce protocole est décomposé en quatre étapes :

1. description générale de l'incident ;
2. identification des points de décision lors de la résolution ;
3. sondage approfondi de ces points ;
4. vérification et validation des éléments rapportés.

Le protocole est appliqué successivement aux différents incidents rapportés par l'opérateur (ou au SME) permettent d'obtenir une information structurée sur les connaissances nécessaires à la prise de décision dans le domaine.

Étape 2 : construction des objets de connaissance

La phase d'extraction d'objets de connaissance (*knowledge objects*) consiste à analyser les transcriptions des entretiens et observations réalisés dans la première phase

afin d'en extraire uniquement les éléments importants, c'est à dire les « concepts détectés, classés et manipulés par l'équipage » (Stanton *et al.*, 2006).

Étape 3a : construction du réseau de connaissance

Les objets de connaissance et leurs relations sont traduits sous forme de réseaux propositionnels construits à partir de triplets du type (SUJET, RELATION, OBJET) issus des transcriptions ou adaptés des observations de l'activité des opérateurs. Ainsi, chaque objet de connaissance est identifié dans l'étape précédente et peut représenter un SUJET ou un OBJET du triplet. Son intégration au réseau propositionnel se traduit par l'insertion d'un nœud. Les propositions sont insérées une à une, chaque couple d'objets (nœuds) étant lié par une relation, visualisée sous la forme d'un lien étiqueté par le mot de la RELATION.

La compilation de l'ensemble des données permet de construire un réseau propositionnel global, représentant l'ensemble des connaissances pertinentes pour la réalisation de la mission par les opérateurs. Afin d'éviter les doublons et autres incohérences, il peut être nécessaire d'adapter ou de fusionner certains triplets issus des entretiens.

La figure 3.10 présente un extrait du réseau propositionnel construit dans le cadre d'une étude sur une salle de commande et de contrôle d'un réseau de distribution d'énergie gérant 341 sous-stations (Salmon *et al.*, 2008b).

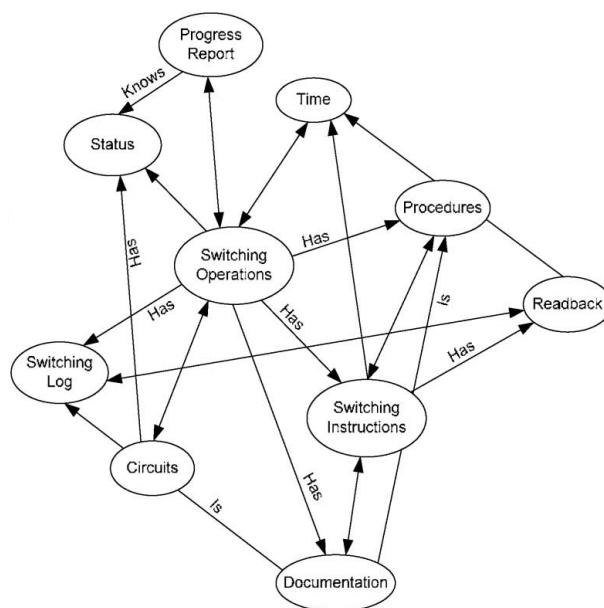


FIGURE 3.10 : extrait de réseau propositionnel construit dans le cadre d'une étude dans le domaine de la distribution d'énergie (Salmon *et al.*, 2008b).

Étape 3b : activation des connaissances

Lorsqu'un réseau propositionnel complet a été construit, il est possible de repré-

senter de manière visuelle, à travers la coloration des nœuds du réseau par exemple, les objets de connaissance activés lors de l'exécution d'une tâche donnée. Ainsi, à partir d'un réseau global représentant l'ensemble des éléments pertinents à la réalisation de la mission, Stanton *et al.* (2006, figures 9 à 15) proposent de construire autant de réseaux propositionnels « coloriés » aux couleurs des rôles / tâches (un réseau par rôle et par tâche), illustrant la distribution de la SA à travers les différents opérateurs et leurs tâches « dans un sens très littéral », exprimant la définition de cette SA distribuée comme « les connaissances activées pour une tâche spécifique d'un système ».

3.3.2.3 Lien avec la théorie des schémas

3 Stanton *et al.* (2009) ancrent le modèle de DSA dans la théorie des schémas, suivant l'idée que chaque opérateur construit sa *Situation Awareness* à partir de schéma prédéfinis, appelés schémas « génotypes ». Ceux-ci sont particularisés à travers l'interaction de l'opérateur avec le reste du système (autres agents humains et technologiques) et la prise en compte des données de la situation (objectifs, environnement, autres variables aléatoires internes ou externes), créant pour chaque situation un schéma « phénotype » particulier. Les entretiens décrits dans l'étape 1 de la méthodologie des réseaux propositionnels permettent, en sondant les opérateurs sur des événements particuliers et leur déroulement, d'obtenir des schémas phénotypes. Chaque entretien donne accès à un schéma particularisé de résolution d'incident par exemple, définissant des « structures de connaissances activées qui guident l'exploration et l'interprétation des informations » lors de la réalisation de la tâche.

Stanton *et al.* (2009) suggèrent qu'avec « assez de schémas phénotypes, il peut être possible d'identifier les invariants dans les performances » des opérateurs, formant ainsi la « base d'un schéma génotype ». C'est ce processus de fusion des schémas phénotypes en un schéma *global* (sous forme de réseau propositionnel) qui est suivi lors de l'étape 3 (3a et 3b) de la méthodologie. Il permet de définir un schéma associé à l'ensemble de la mission pour le système donné. La visualisation des objets de connaissance sous forme d'un réseau propositionnel et de leur activation lors de la réalisation d'une tâche (à travers leur coloration) permet d'accéder à ce schéma génotype, particularisé par l'exécution des tâches.

Le théorie de DSA proposée par Stanton *et al.* (2009) permet de décrire la SA comme une composante émergente du système et le modèle associé, à la fois descriptif par ses fondements théoriques reconnus (cognition distribuée, théorie des schémas, cycle perception-action) et prescriptif par sa concrétisation à travers la construction de réseaux propositionnels, offre une alternative intéressante au courant dominant de la *Team SA*, souvent focalisée sur l'aspect « humain » de l'interaction homme-machine.

3.4 MESURER LA SITUATION AWARENESS

La multitude de définitions de la SA existantes (§ 3.1) et de modèles associés (§ 3.2) introduit de nombreuses méthodes de mesures de ce concept, en particulier dans le domaine de la SA individuelle. Comme d'autres concepts en sciences cognitives (nous pensons à celui de charge de travail notamment), le concept de *Situation Awareness* semble difficile à quantifier, en particulier en temps réel.

Selon Endsley (1995a), trois critères sont nécessaires afin d'établir la validité et fiabilité d'une méthode de mesure de la SA.

- Il faut s'assurer que la méthode mesure uniquement les composantes qu'elle annonce mesurer et non le résultat d'autres processus ;
- les informations collectées doivent être suffisamment détaillées et compréhensibles ;
- la mesure ne doit pas influencer les autres processus en cours lors de la réalisation de la tâche, afin de ne pas biaiser les données et d'éviter d'induire un changement de comportement de la part de l'opérateur.

Nous proposons dans un premier temps de détailler une classification des techniques de mesure proposée par Endsley (1995a), illustrée par quelques exemples de méthodes mises en œuvre pour mesurer la SA. Dans un second temps, nous reviendrons en détail sur certaines de ces méthodes.

3.4.1 Revue des techniques de mesure de la Situation Awareness

Endsley (1995a) différencie d'une part les méthodes de mesure de la *Situation Awareness* objectives et d'autre part les méthodes subjectives.

3.4.1.1 Techniques objectives de mesure de la SA

Les méthodes « objectives » reposent sur des mesures quantifiables de différents paramètres concrets à travers des capteurs (mesures physiques, performance) et définis préalablement à l'expérimentation.

Les mesures physiologiques sont effectuées directement sur les opérateurs, à travers des capteurs posés sur le corps (électroencéphalogramme) ou installés à proximité (oculométrie).

L'interprétation de la mesure des ondes émises par le cerveau (à travers un électroencéphalogramme) peut s'avérer difficile, celles-ci ne donnent pas d'informations précises sur le fait que les informations perçues ont été stockées en mémoire ou sur la façon dont l'information a été comprise par le sujet.

Les mesures de type *eye-tracking* (oculométrie) permettent d'acquérir des informations sur le regard de l'opérateur lors de l'exécution de sa tâche. Néanmoins, l'observation d'un élément ne garantit pas son traitement cognitif ultérieur et ainsi la construction de la *Situation Awareness* associée à ce sujet.

Les mesures de performance consistent à mesurer les résultats des tâches de l'opérateur sur l'interface et leur adéquation avec les objectifs fixés. Ces mesures sont divisées en plusieurs catégories :

- Les *mesures globales* de performance permettent de mesurer de manière non intrusive le résultat général d'un opérateur pour une tâche précise. A ce titre, il est difficile de relier ce type de mesure à celle de la *Situation Awareness*. En effet, un mauvais résultat de l'opérateur peut résulter de différents facteurs et non uniquement du fait que sa *Situation Awareness* soit mauvaise : cette mesure traduit une performance liée à l'ensemble de la chaîne de processus, depuis l'accès à l'information jusqu'à la prise de décision.
- Les *mesures impliquant des tâches externes* consistent à supprimer ou modifier des informations de l'interface pendant l'expérimentation et à mesurer le temps de réaction de l'opérateur vis-à-vis de ces modifications. Cette méthode, en plus d'être intrusive (le sujet cherche à comprendre pourquoi les informations manquent), ne permet pas d'être entièrement certain que les changements ont été perçus. En effet, l'opérateur compensera parfois ces changements en essayant de deviner leur cause (mal-fonction de l'appareil, coupure volontaire de communication...) et ne fera pas de remarque spécifique à ce sujet lors du débriefing.
- Les *mesures sur les sous-tâches* de la tâche principale consistent à mesurer la performance de certaines parties de la mission de l'opérateur. Ces mesures non-intrusives donnent des informations plus précises qu'une mesure globale mais uniquement sur la partie de la SA concernant la sous-tâche mesurée. Cette technique permet d'acquérir des informations de la SA « locale », propre à une sous-tâche et ne permet pas de vérifier la dépendance des différents composants du système : augmenter la SA pour une sous-tâche peut réduire celle, non mesurée, d'une autre sous-tâche. De plus, si le sujet comprend sur quels critères sa performance est mesurée, un biais peut être introduit, le sujet se focalisant sur certains éléments et certaines sous-tâches au détriment des autres, entraînant une évaluation erronée de la SA.

3.4.1.2 Techniques subjectives de mesure de la SA

Les mesures « subjectives » reposent sur l'évaluation, par le sujet ou un observateur tiers, de la SA du sujet, à travers différentes techniques.

L'auto-estimation consiste à laisser l'opérateur s'auto-évaluer sur son niveau estimé de *Situation Awareness*. L'évaluation, généralement effectuée sur une (ou plusieurs) échelle de Likert, est demandée à l'opérateur après qu'il ait exécuté la tâche considérée ; elle est proche d'une évaluation de type NASA-TLX pour la charge de travail (Hart et Staveland, 1988). Cette méthode, non-intrusive, est très peu coûteuse en terme de développement, puisque réalisable facilement à l'aide d'un papier et d'un crayon. Parmi les méthodes d'auto-estimation (*self rating technique*), nous retenons notamment CARS (McGuinness et Foy, 2000)

et SART (Taylor, 1990), cette dernière étant la plus répandue (§ 3.4.2.2).

Il a été montré que les résultats des évaluations avec SART sont corrélés avec la performance de l'opérateur dans la tâche exécutée : un opérateur ayant réussi sa tâche s'attribuera un meilleur score de SA qu'un opérateur qui l'aurait raté (Selcon et Taylor (1990) cité par Endsley (1995a)). Endsley (1995a) suggère en conséquence que ce type de mesure correspond plutôt à un score de « confiance en terme de SA, auto-attribué par l'opérateur ».

L'estimation par un observateur extérieur consiste à faire intervenir une personne tierce pour évaluer la *Situation Awareness* du sujet. Cet observateur doit connaître la situation et le système parfaitement. Le recueil se fait soit par l'observation directe (assez limitée, l'observateur ne peut reporter que ce qu'il voit à l'écran, et éventuellement les réactions du sujet) ou à travers la verbalisation du sujet.

Afin de limiter au maximum l'impact de la présence d'un observateur (environnement « laboratoire »), celui-ci peut jouer le rôle d'un tiers (contrôleur aérien ou co-pilote par exemple), posant des questions sur la situation comme en situation réelle (voir la méthode SPAM (Durso *et al.*, 1995), § 3.4.2.3). Cette technique ne permet néanmoins pas de s'assurer que toutes les informations sont connues (il faudrait constamment faire des *check-lists* de ce que l'opérateur a en mémoire) et peut introduire un biais en focalisant le sujet sur les éléments de situation évalués : par exemple une question sur l'altitude répétée trop souvent portera l'attention de l'opérateur sur ce paramètre.

3.4.1.3 Les questionnaires

A cheval entre les méthodes subjectives et objectives, les questionnaires permettent de recueillir des données précises et objectives sur la situation (question du type « quel est le statut de tel objet ? ») mais le moment du recueil peut influencer sur les résultats. Différentes méthodes de questionnaires existent, chacune ayant ses avantages et inconvénients :

Les questionnaires post-test sont complétés par l'opérateur à la fin d'un scénario, revenant sur différents instants clés du scénario. Cela laisse à l'opérateur tout le temps pour réfléchir aux réponses qu'il donnera sur les éléments évalués. L'inconvénient est que certains événements importants peuvent être oubliés après l'expérimentation alors que l'opérateur les a pris en compte et compris dans la réalisation de sa mission. De plus, Endsley (1995a) remarque que les opérateurs ont tendance à généraliser et tentent de rationaliser a posteriori les événements qui sont intervenus lors de l'expérimentation. Cette interprétation, à l'inverse d'un simple compte-rendu, peut induire de faux résultats sur l'état de la *Situation Awareness* de l'opérateur lors de la mission.

Les questionnaires en ligne consistent à poser des questions à l'opérateur au cours de l'expérimentation. Ces questions, considérées comme une tâche secondaire et donc intrusive, peuvent augmenter la charge de travail de l'opérateur (souvent déjà élevée) ou impacter les performances globales de la tâche principale, parfois de manière positive en attirant l'attention de l'opérateur sur un élément qu'il n'avait pas pris en compte, améliorant ainsi sa *Situation Awareness*. Mesurer le temps de réponse dans de tels cas n'est pas pertinent puisque celui-ci dépend de la stratégie de gestion du temps de chaque opérateur : certains préfèrent répondre directement, aux questions, d'autres attendent d'avoir terminé leur tâche courante.

Les questionnaires de type *freeze* « gèlent » l'expérimentation afin de poser une question à l'opérateur sur un élément de la situation non connu à l'avance et sélectionné de manière aléatoire (afin d'éviter de porter son attention sur certains éléments en particulier) et à des temps aléatoires de l'expérimentation (afin d'éviter l'effet de préparation). Lorsque la question apparaît, le reste de l'écran est généralement masqué. L'outil SAGAT (Endsley, 1995a, 2000b), détaillé dans la partie 3.4.2.1, correspond à cette technique de questionnaire.

3.4.2 Quelques méthodes de mesure de la SA

Plusieurs auteurs ont proposé des revues et comparaisons de plusieurs méthodes de mesures de la *Situation Awareness*. Jeannot *et al.* (2003, tableau 4) proposent une comparaison de neuf méthodes de mesure et notamment de leur applicabilité aux systèmes de contrôle aérien. Nous retrouvons une partie de ces méthodes dans une analyse de Salmon *et al.* (2006, tableau 3) qui comparent 17 méthodes et leur applicabilité au domaine C4I (*Command, Control, Communications, Computers, Intelligence*).

Notons que le contexte dans lequel s'effectue la mesure est important car il définit notamment son intrusivité dans l'activité de l'opérateur et son adaptabilité à un cadre non expérimental. Par exemple, les méthodes de type *freeze* sont difficilement applicables en situation réelle, où l'activité de l'opérateur ne peut être interrompue (difficile d'imaginer l'arrêt en vol d'une activité de pilotage!) et sont donc généralement menées sur simulateur. Nous présentons dans la suite de cette section les méthodes de mesure les plus souvent citées et utilisées dans la littérature.

3.4.2.1 SAGAT (*Situation Awareness Global Assessment Technique*)

La méthode SAGAT (*Situation Awareness Global Assessment Technique*) (Endsley, 1995a, 2000b) est la méthode d'évaluation de la *Situation Awareness* la plus répandue dans la littérature. Il s'agit d'un questionnaire de type *freeze* : lors du déroulement de l'expérimentation, l'écran de l'opérateur est gelé, les interfaces sont masquées, et plusieurs questions sur l'état de la situation sont posées à l'opérateur.

Les requêtes automatisées, déclenchées plusieurs fois par scénario, n'ont pas d'effet sur la performance globale de l'opérateur, même après la reprise (Endsley, 1995a). L'intervalle entre chaque requête est au minimum de une minute et la première requête ne doit pas être déclenchée moins de trois minutes après le début du scénario afin de permettre au sujet de se construire une image mentale de la situation. Endsley (1995a) conseille également de limiter la durée de chaque gel à environ deux minutes, quitte à ne pas poser l'ensemble des questions prévues pour ce *freeze* par le système. Les intervalles entre les différents gels et les questions posées pendant chaque gel varient d'une requête à l'autre afin d'éviter tout biais lié à un effet de préparation.

Cette technique d'évaluation de la SA a été appliquée dans plusieurs domaines tels que le contrôle aérien (Endsley *et al.*, 2000b) ou le pilotage d'avions de chasse (Endsley, 2000a). Notons également que cette méthode a été utilisée dans le cadre d'étude sur la SA d'équipe, afin de pouvoir mesurer les SA individuelles et de les comparer (Saner *et al.*, 2009). Les éléments de situation sur lesquels l'opérateur sera interrogé lors des *freeze* peuvent être issus de l'analyse préalable et la modélisation des exigences, telle que présentée dans le paragraphe 3.2.1.1. Nous remarquons qu'outre le fait qu'elle soit intrusive, cette méthode ne peut être implémentée que sur un simulateur : il n'est pas possible, en situation réelle, de geler la situation pour évaluer la SA de l'opérateur.

3.4.2.2 SART (*Situational Awareness Rating Technique*)

La méthode SART (*Situational Awareness Rating Technique*) (Taylor, 1990) a également été beaucoup utilisée. Dans ce questionnaire à trois dimensions, le sujet évalue sur une échelle à 7 niveaux, sa compréhension de la situation lors de l'exercice qu'il vient de terminer. Les trois dimensions évaluées sont la « demande de ressources attentionnelles » (*Demand on attentional resources*), la « fourniture de ressource attentionnelle » (*Supply of attentional resources*) et la « compréhension » (*Understanding*). A partir de ces trois dimensions, un calcul d'une valeur « globale » de la SA peut être obtenu par la formule :

$$SA = Understanding - (Demand - Supply)$$

Une version de cette technique propose un questionnaire en 10 dimensions (Taylor, 1990), traduisant les 3 dimensions citées à des niveaux de précisions plus « fins » (intégrant des dimensions telles que l'« instabilité de la situation », sa « complexité », ou l'« éveil de l'opérateur » par exemple). SART offre l'avantage de ne pas être intrusive. Néanmoins, elle est critiquée dans la littérature pour sa généralité (elle ne permet pas d'obtenir de détails sur les éléments de situation problématique par exemple) et sa subjectivité.

3.4.2.3 SPAM (*Situation Present Assessment Method*)

La méthode SPAM (*Situation Present Assessment Method*), proposée par Durso *et al.* (1995), s'appuie sur un observateur extérieur afin de mesurer à la fois la charge de travail et la SA d'un opérateur (un contrôleur aérien dans le cadre de l'étude de Durso *et al.* (1995, 1998) par l'intermédiaire de questions directement posées à l'opérateur.

Dans le paradigme expérimental présenté, le contrôleur aérien est appelé sur sa ligne téléphonique (situation réelle d'activité), et deux mesures sont effectuées :

- le temps qu'il met à décrocher le téléphone mesure la charge de travail ;
- le temps mis pour répondre correctement à chaque question posée à l'opérateur par le tiers indique la SA de l'opérateur.

Nous remarquons qu'en utilisant cette méthode, la mesure de SA (temps de réponse) est dépendante du fait que l'opérateur connaît la réponse ou qu'il sait comment trouver cette information rapidement.

Durso et Dattel (2004) présentent également une étude permettant de mesurer la SA d'un conducteur de voiture en situation réelle, à l'inverse de l'étude sur les contrôleurs qui se déroule en laboratoire, sur simulateur. Le même schéma en deux temps est mis en place : le passager avant de la voiture demande au conducteur s'il est prêt à répondre à une question (mesure de la charge de travail) et, le cas échéant, pose la question (affichée sur un ordinateur portable) et en mesure le temps de réponse. Un deuxième passager installé à l'arrière note si la réponse du conducteur est correcte.

3.4.2.4 SASHA

La méthode SASHA (pour *SA for SHAPE*, le projet SHAPE (*Solutions for Human-Automation Partnerships in European ATM*) vise à améliorer l'automatisation des stations de contrôle aérien en Europe) est basée sur les avantages de plusieurs méthodes présentées précédemment (Jeannot *et al.*, 2003) et se décline en deux versions : une version *en ligne* (SASHA-L) et une version *questionnaire* (SASHA-Q). Jeannot *et al.* (2003) indiquent que cette technique d'évaluation a été développée pour pallier le fait que SAGAT est trop intrusif et SART trop peu détaillée.

La version *en ligne*, **SASHA-L**, est une amélioration de la méthode SPAM (§ 3.4.2.3). L'expert a accès à la situation en temps réel à travers une recopie d'écran du sujet et peut poser des questions sur la situation et les outils quand il le désire (à l'opposé de SPAM où les questions sont prédéfinies par la trame du scénario). Le temps de réponse n'est plus directement mesuré mais est évalué par l'expert : celui-ci choisit de le noter comme « OK », « trop court » (qui pourrait indiquer une réponse trouvée de façon aléatoire) ou « trop long ». L'expert définit également un niveau d'importance pour chaque question posée.

Une formation à cette technique est ainsi nécessaire afin d'avoir le même type d'évaluation quelque soit l'expert mais il est difficile d'homogénéiser les évaluations des experts : la variété des questions et leur catégorisation par l'expert introduit une part de subjectivité dans le processus d'évaluation.

Le questionnaire associé à cette technique, **SASHA-Q**, propose un ensemble de questions basées sur la littérature du domaine des facteurs humains ou spécifiques à la simulation, pour lesquelles l'opérateur doit répondre à la fois sur une échelle à 5 niveaux et par un champ libre de commentaire.

Bien que construite pour le domaine du contrôle aérien et donc très orientée vers cette littérature et les problématiques liées à cette activité, cette méthode présente une approche écologique (au sens de Gibson (1979)) s'intégrant au scénario sans perturber l'opérateur (à la manière de SPAM). Néanmoins, comme pour ces méthodes de sondage « en ligne », on peut craindre que les questions posées à l'opérateur dirigent son attention vers les éléments de situations sondés, et donc induisent un biais dans la mesure de SA pour les questions suivantes. Nous n'avons pu trouver, dans la littérature, de validation ou d'autre application de la méthode SASHA.

3.4.3 Commentaires concernant la mesure de la SA

Dans cette section, nous avons présenté un aperçu de différentes méthodes de mesure de la *Situation Awareness*. Proposées il y a plus de 15 ans, les méthodes SAGAT et SART restent les deux méthodes les plus souvent utilisées et citées dans la littérature.

D'autres techniques de mesures existent, Jeannot *et al.* (2003) ainsi que Salmon *et al.* (2006) en décrivent certaines comme par exemple SAVANT (intégrant SAGAT et SPAM), SALSA (une technique de type *freeze*, comme SAGAT), SAPS (basée sur des requêtes « en ligne »), C-SAS (un ensemble d'échelles d'évaluation à différents instants de l'expérimentation), SALIANT (réponses comportementale du sujet face à des comportements pré-définis) ou SA/BARS (échelles d'évaluation). Nous notons que ces méthodes ont été conçues pour, et donc répondent, à différents besoins opérationnels (SASHA est par exemple très orientée *contrôle aérien*).

Certaines études montrent d'ailleurs que les éléments mesurés par ces différentes méthodes ne sont pas forcément les mêmes. Une comparaison de SAGAT et SART par Endsley *et al.* (1998b) ne détecte pas de corrélation entre les résultats des deux mesures, et indique que l'évaluation, subjective, de la SA dans le questionnaire SART, est corrélée à la mesure de performance. L'absence de corrélation entre SAGAT et SART a également été montrée par Salmon *et al.* (2009). A l'inverse, dans un autre article, Endsley *et al.* (2000b) notent une corrélation entre les mesures de SART et celles de SAGAT relatives aux questions de niveau 1 de la *Situation Awareness*. Par contre, dans cette étude, les auteurs indiquent qu'il n'y a pas de différence

significative sur le score SART lorsque l'opérateur utilise des interfaces différentes (permettant d'acquérir une SA plus ou moins bonne), contrairement à SAGAT.

Ces résultats parfois contradictoires peuvent amener à se questionner sur la validité de certaines méthodes de mesure de la *Situation Awareness*. Mais l'objet des mesures et la méthodologie employée étant différents (gel de l'écran dans un cas, questionnaire post-exercice dans l'autre), la difficulté pour le concepteur réside dans le choix de la méthode selon le modèle de SA sélectionné et les objectifs définis pour l'utilisation de cette méthode (évaluation spécifique de l'interface ou de la SA « globale » de l'opérateur, ...).

3.5 SYNTHÈSE DU CHAPITRE 3

A travers les quatre sections de ce chapitre, nous avons cherché à comprendre comment pouvaient être décrites les informations nécessaires à l'opérateur dans la réalisation des différentes tâches de sa mission. En psychologie cognitive et en facteurs humains, ce concept est exprimé par le terme de *Situation Awareness*.

- La première partie (section 3.1) a permis d'introduire ce terme et d'en donner plusieurs définitions, correspondant à diverses approches, se contredisant parfois, et décrivant notamment ce concept comme un état de connaissance ou le processus d'acquisition de cet état.
- La deuxième partie (section 3.2) décrit quelques modèles, classés dans deux catégories : descriptifs et prescriptifs, s'appuyant sur les définitions de la section précédente.
- La troisième partie (section 3.3) a permis d'aborder le concept de *Situation Awareness* sous l'angle des systèmes collaboratifs, où, d'un point de vue du système, plusieurs tâches sont exécutées, chacune nécessitant d'adopter un point de vue différent sur la situation, de la même manière qu'un opérateur unique d'un système complexe devra adopter plusieurs points de vue différents au cours de sa mission pour en réaliser les différentes tâches qui s'enchaînent.
- Enfin, la dernière partie (section 3.4) nous a permis de présenter quelques méthodes de mesure de la *Situation Awareness*, et de mettre en exergue l'importance de la compréhension de l'objet de la mesure dans le choix de celle-ci. Ce choix devant être effectué en lien avec le modèle de *Situation Awareness* sélectionné.

A l'inverse du chapitre précédent, où le bilan « informationnel » des situations associées aux tâches décrites n'était pas formalisé ou même pris en compte dans l'analyse, les travaux présentés dans ce chapitre proposent différentes solutions permettant de formaliser ces données à travers la définition du concept de *Situation Awareness* et de modèles le soutenant.

Néanmoins, nous remarquons que les aspects changeants de ces situations, induits par les transitions entre les tâches rencontrées au cours d'une mission, ne sont

pas directement considérés dans les études présentées, qui dressent l'inventaire des exigences en termes de SA nécessaires à la réalisation de la mission de l'opérateur dans sa globalité, et non pour chaque tâche exécutée séparément. Seuls les travaux sur les systèmes collaboratifs, où les opérateurs sont en charge de tâches peut-être plus spécialisées, proposent cette description plus fine de la SA (sous forme d'un vecteur d'éléments de situation associé à chaque opérateur pour Saner *et al.* (2009), ou d'un réseau propositionnel coloré dans les travaux de Stanton *et al.* (2009)).

Dans ces travaux, il manque ainsi le lien entre la description de la dynamique temporelle des tâches et la dynamique de la situation associée : les exigences en termes de *Situation Awareness* changent en fonction de la tâche réalisée par l'opérateur. Afin de combler ce manque, nous proposons dans la suite de notre étude une nouvelle approche permettant d'intégrer à la fois la description des transitions entre tâches et celle des exigences de *Situation Awareness* associées à chacune d'entre elles.



DEUXIÈME PARTIE : DE LA MODÉLISATION À L'ANALYSE DES TRANSITIONS, CONTRIBUTION CONCEPTUELLE

4

Les transitions et la dualité tâche / contexte

L'état de l'art présenté dans les chapitres 2 et 3 a mis en avant l'absence de lien formel entre la description des tâches, en particulier dans leur dynamique temporelle, et les contextes de situation associés à chacune de ces tâches. Or, l'objectif de cette thèse étant de permettre l'analyse des transitions entre tâches et plus particulièrement de leur impact sur la *Situation Awareness* de l'opérateur, il nous semble important de formaliser ce lien entre la description de la dynamique des tâches et celle de la *Situation Awareness* construite par l'opérateur pour la réalisation de ces tâches.

Nous proposons ainsi dans ce chapitre d'adopter une approche duale dans la description de l'activité des opérateurs, afin de définir, pour chaque tâche exécutée au cours d'une mission, les éléments de *Situation Awareness* exigés (au sens anglais de « *requirements* ») par l'opérateur pour atteindre une performance satisfaisante dans la réalisation de cette tâche. Cet *ensemble* d'éléments contribuant à la construction de la *Situation Awareness* de l'opérateur, spécifié pour chacune des tâches considérées dans la description de l'activité, constitue ce que nous appelons le « *contexte* » de la tâche.

Dans ce chapitre, nous nous appuyons sur cette dualité tâche / contexte afin de proposer, dans une première partie (section 4.1), une taxinomie des transitions entre tâches. Nous présenterons ensuite (section 4.2) une adaptation du modèle de *Distributed Situation Awareness* (DSA) proposé par Stanton *et al.* (2006) que nous appliquons aux systèmes complexes centrés sur un opérateur unique, afin de modéliser les exigences *Situation Awareness* définis pour chaque tâche de ce système, exposant ainsi un premier modèle de transitions. Enfin, nous présenterons dans la section 4.3 la façon dont nous envisageons d'appliquer cette taxinomie et le modèle de transitions proposés dans les sections 4.1 et 4.2.

Dans ce chapitre, nous considérons que les tâches et éléments de situation nécessaires à leur exécution (le « contexte » de la tâche) sont connus. Nous proposons dans le chapitre suivant une méthodologie détaillant, entre autre, la façon dont peuvent être extraites, modélisées et analysées ces données.

4.1 UNE PROPOSITION DE TAXINOMIE DES TRANSITIONS

Face à l'absence de cadre de travail global pour l'analyse des transitions, nous souhaitons proposer dans cette section une première piste pour cette analyse. Décrit sous la forme d'une taxinomie, ce cadre de travail a pour objectif de répertorier et de qualifier a priori les différents types de transitions que l'on pourra rencontrer lors de la description d'un système complexe. Cette taxinomie catégorise les différents types de transitions selon deux axes, correspondant aux deux axes explorés dans notre état de l'art :

- l'axe **temporel** décrit l'enchaînement temporel des tâches de l'opérateur.
- l'axe **contextuel** décrit l'évolution du contexte de la situation lors du changement de tâches.

Ainsi, dans la taxinomie, une transition sera caractérisée par un point du plan référencé par ces deux axes. Nous proposons dans la suite de cette section d'étudier successivement la façon dont nous pouvons caractériser une transition sur chacun de ces axes. La taxinomie des transitions que nous présentons sera synthétisée dans le tableau 4.2, illustrant l'ensemble des transitions identifiées.

Avant de détailler les dimensions de cette taxinomie, nous souhaitons présenter les travaux de Allen (1983) qui définit de manière exhaustive les relations entre les intervalles temporels.

4.1.1 La logique des intervalles temporels de Allen

Dans ses travaux, Allen (1983) identifie 5 relations temporelles possibles entre deux intervalles de temps – décrits à partir de leurs points de départ et d'arrivée – qui sont l'« anachronisme », la « séquence », la « concomitance », la « coïncidence » et le « parallélisme » (nommées d'après les travaux de Vernier et Nigay (2001)). Elles définissent cinq relations « prototypes » à partir desquelles d'autres relations vont pouvoir être construites (tableau 4.1, première colonne « prototype »).



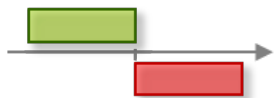





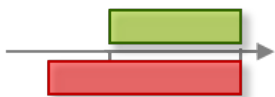
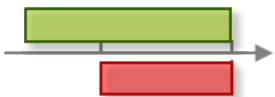
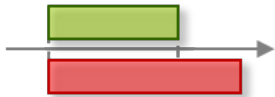
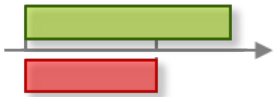

Ainsi, afin d'apporter davantage de précisions sur l'enchaînement des deux intervalles considérés, Allen définit 13 manières différentes de décrire formellement l'agencement entre deux intervalles de temps. Ces 13 relations sont construites à partir des symétriques de chacune des 5 relations « de base », sauf le parallélisme, et en décomposant la relation de coïncidence en 3 relations possibles (intervalles coïncidant au début, au milieu ou à la fin). L'ensemble de ces treize relations est présenté dans le tableau 4.1.

Les relations proposées par Allen ont été appliquées dans de nombreux domaines, et le formalisme mathématique apporté initialement les rend particulièrement utiles dans le cadre de raisonnements automatisés, appuyés par la table de transitivité proposée par Allen (1983).

Ces relations ont également été utilisées par Vernier et Nigay (2001) afin de

CHAPITRE 4. LES TRANSITIONS ET LA DUALITÉ TÂCHE / CONTEXTE

TABLEAU 4.1 : les relations de Allen (1983), familles nommées par Vernier et Nigay (2001).

Familles	Prototypes	Variantes
Anachronisme	 X avant Y	 Y avant X
Séquence	 X rencontre Y	 Y rencontre X
Concomitance	 X recouvre Y	 Y recouvre X
Coincidence	 X pendant Y	 Y pendant X
	 X termine Y	 Y termine X
	 X commence Y	 Y commence X
Parallélisme	 X égal à Y	

définir un cadre de travail pour la caractérisation des modalités de sortie d'un système multimodal ; les cinq relations prototypes sont alors combinées avec quatre aspects de combinaison (temporel, spatial, syntactique et sémantique). Plus récemment, Jourde (2011) a intégré ces relations sous la forme d'opérateurs binaires pour la description des relations entre tâches « sœurs » d'un arbre de tâche dans la notation COMM (COllaboratif et MultiModal, voir aussi §2.1.2.7). L'auteur précise que le niveau de précision induit par ce type de relations n'est pertinent que pour la description de tâches de niveau d'abstraction faible, correspondant aux couches basses de l'arbre (description de la combinaison de modalités par exemple).

4.1.2 Les deux dimensions de la taxinomie

Dans cette section, nous décrivons les deux dimensions retenues pour la caractérisation des transitions, en s'appuyant notamment sur les relations de Allen que nous adapterons de façon pertinente au niveau d'abstraction considéré dans notre étude afin de décrire l'évolution temporelle et contextuelle des tâches.

4.1.2.1 Description des transitions sur l'axe temporel

Le temps étant un élément clé dans l'étude des systèmes dynamiques, la projection des transitions sur un axe temporel semble naturelle. Nous basons notre description sur une version simplifiée des relations de Allen (1983).

Fondamentalement, l'ensemble de ces relations peut se résumer à deux groupes de relations temporelles :

- le groupe des **relations séquentielles**, constitué des 2 relations de la famille des *anachronismes* et *séquences* (soit 4 relations au total en incluant les relations symétriques). Les intervalles considérés dans ce groupe sont entièrement disjoints d'un point de vue temporel (dans le cas de la relation de « rencontre », la fin d'un intervalle marque le début du suivant).
- le groupe des **relations parallèles** (ou simultanées), constitué des 3 autres familles : *concomitance* (2 relations), *coïncidence* (6 relations) et *parallélisme*. Les intervalles de ce groupe ont une partie commune, où les intervalles sont superposés.

L'agencement temporel des tâches du système peut être identifié à celui des intervalles. Ainsi, par identification, les deux groupes de relations entre intervalles définis précédemment correspondent à deux types de transitions entre tâches. Nous proposons les définitions de transition entre deux tâches T_1 et T_2 suivantes :

- Nous appelons **transition simple** le changement de tâches entre 2 tâches temporellement disjointes (relation de type *anachronisme* ou *séquence*) : lors de ces transitions, la tâche initiale T_1 se termine et est suivie entièrement par une tâche T_2 , sans recouvrement temporel entre les deux tâches.
- Nous appelons **ajout de tâche** le changement de tâches où les 2 tâches partagent, temporellement parlant, une partie de leur exécution : l'ajout de tâche

correspond au moment où la tâche T_2 est ajoutée à la tâche T_1 .

Afin de compléter ces deux catégories, nous ajoutons à cette dimension temporelle un troisième type de transitions, très présent dans la littérature : les **interruptions**. Comme décrite dans la section 2.3 de ce mémoire, une interruption correspond à l'enchaînement en séquence d'une tâche T_1 et d'une tâche T_2 , suivi de la reprise de la tâche T_1 . Cette nouvelle catégorie constitue ainsi une combinaison de deux *transitions simples*. Nous pensons qu'il est nécessaire de faire apparaître cette catégorie de façon séparée dans cette taxinomie dont le but est de permettre aux concepteurs et experts du domaine de classer les transitions identifiées lors de l'analyse de l'activité.

L'ajout de ce type de transition est justifié par la littérature. En effet, l'état de l'art sur les interruptions (section 2.3) nous permet de penser que les conséquences en termes de conception d'interfaces seront différentes pour la prise en compte d'une transition simple ou celle d'une interruption par une tâche secondaire. Notamment, lorsque l'on conçoit une interface pour une interruption, il est nécessaire d'intégrer à l'avance le concept de reprise de la tâche initiale tandis que dans un changement de tâches simple, la tâche T_1 ne sera plus abordée pendant un certain laps de temps.

Ainsi, dans cette section nous avons défini 3 catégories pour décrire l'aspect temporel des transitions entre tâches : les **transitions simples**, les **ajouts de tâche** et les **interruptions**. Chacune de ces catégories peut être croisée avec la description de la transition sur l'axe contextuel.

4.1.2.2 Description des transitions sur l'axe contextuel

La deuxième dimension de notre approche d'analyse des transitions concerne l'aspect contextuel de ces transitions. Le contexte d'une tâche définit l'*ensemble* des exigences en termes de *Situation Awareness* associées à cette tâche. Afin de représenter ces contextes et leur évolution au fil des tâches, nous utilisons donc naturellement la notation ensembliste et, en particulier, les diagrammes de Venn.

Les relations de Allen (1983) ont été utilisées à d'autres fins que la description de relations entre intervalles temporels, par exemple pour la description de la relation spatiale entre différents éléments abstraits d'interface (Limbourg *et al.*, 2004). Nous proposons d'adapter ces relations à notre approche ensembliste, en considérant les relations entre des paires d'ensembles d'éléments de situations, correspondant aux contextes de différentes tâches étudiées. Ainsi, nous pouvons adapter les 13 relations présentées dans le tableau 4.1 en raisonnant en terme d'intersections d'ensembles :

- Les 4 relations d'*anachronisme* et de *séquence* peuvent être rassemblées en une seule relation de **séparation**, où les deux contextes sont disjoints (aucun élément de situation n'est partagé entre les deux contextes) ;
- Les 2 relations de *concomitance* peuvent être rassemblées en une seule relation

d'**intersection**, où les deux contextes partagent un sous-ensemble d'éléments de situation ;

- La relation de *parallélisme* est conservée, nous la nommons « **redondance totale** » afin d'éviter toute confusion avec son équivalent temporel. Les deux contextes partagent exactement les mêmes éléments de situation ;
- Enfin, nous réduisons les 6 relations de *coïncidence* à 2 relations. En effet, d'un point de vue ensembliste, il n'y a pas de différence entre les variantes où les intervalles coïncident au début, au milieu ou à la fin. En transposant les deux relations « X pendant Y » et « Y pendant X », nous obtenons deux relations entre le contexte associé à la première tâche (C_1) et celui associé à la deuxième tâche (C_2)
 - Lorsque C_1 est inclus dans C_2 , nous avons une relation d'**extension**, les éléments de situation de C_1 sont tous dans C_2 , qui en possède d'autres supplémentaires ;
 - Lorsque C_1 inclut C_2 , il s'agit d'une relation de **réduction**, le changement de contexte se focalise sur un sous-ensemble C_2 des éléments de C_1 .

Nous avons ainsi défini cinq relations entre les contextes associés aux tâches de notre transition. Ces 5 relations sont synthétisées sur la figure 4.1. Les noms proposés sont inspirés de ceux proposés par Vernier et Nigay (2001) dans le contexte de combinaison multimodale.

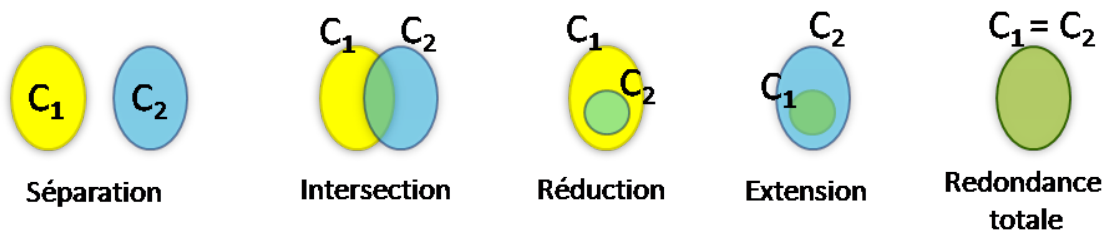


FIGURE 4.1 : cinq relations ensemblistes de la dimension contextuelle de notre taxinomie.

4.1.3 Description de la taxinomie

Dans tout cette section, nous noterons T_1 et T_2 les deux tâches constituant les transitions étudiées, T_1 étant la première tâche à apparaître, précédant T_2 .

Nous proposons dans cette section de détailler chacun de ces types de transition en les illustrant par des exemples d'application aux systèmes complexes lorsque cela est pertinent. Afin de structurer cet exposé, nous le présentons sous l'angle des 3 catégories temporelles identifiées : transitions simples, ajout de tâche et interruption.

4.1.3.1 Transitions simples ($T_1 \Rightarrow T_2$)

Ce paragraphe présente les 5 transitions simples de la taxinomie. Pour ces transitions, T_1 et T_2 sont totalement disjointes d'un point de vue temporel : elles se déroulent l'une après l'autre, avec éventuellement un court laps de temps séparant les deux, mais ne permettant pas l'exécution d'une autre tâche.

Séparation des contextes

Dans cette transition (figure 4.2), les deux tâches sont « totalement » séparées, autant du point de vue temporel que contextuel. Ce changement de tâches nécessite que l'opérateur reconstruise une *Situation Awareness* pour la réalisation d'une deuxième tâche qui ne fait appel à aucun élément de situation commun avec la première.

Il traduit un changement radical pour l'opérateur ; dans notre contexte d'étude, où les opérateurs sont en charge d'activités spécifiques et spécialisées, il sera probablement peu rencontré (à la différence de l'interruption en séparation de contexte, voir §4.1.3.3)

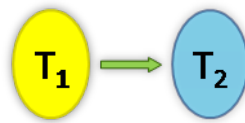


FIGURE 4.2 : transition simple – Séparation des contextes.

Intersection des contextes

Cette transition (figure 4.3) représente un cas de figure courant dans l'activité des opérateurs de systèmes dynamiques, où les changements de tâches nécessitent généralement de conserver une certaine connaissance sur une partie commune d'éléments de situation, partagés entre les deux tâches en jeu.

Nous pouvons illustrer ce cas de figure par la tâche d'un pilote de drone contrôlant deux drones sur la même zone tactique : passer du pilotage d'un drone à l'autre nécessite des interactions disjointes dans le temps mais fait appel à des connaissances partagées entre les deux tâches sur la situation tactique.

Réduction de contexte

Le changement de contexte de cette transition (illustré en figure 4.4) correspond à la restriction du contexte de la tâche courante de l'opérateur à une tâche plus précise (par exemple dans le cas d'une sous-tâche de la tâche initiale) qui se focalise sur une partie du contexte initial.

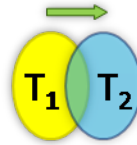


FIGURE 4.3 : transition simple – Intersection des contextes.

Par exemple, il pourrait correspondre, dans le cas de la surveillance maritime, à la tâche d'assistance d'un garde-côte à un navire particulier pour la navigation dans une zone à risque, alors que sa tâche initiale était de surveiller un chenal où navigue un ensemble de bâtiments.

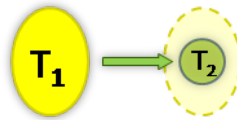


FIGURE 4.4 : transition simple – Réduction de contexte.

Extension de contexte

Cette transition (figure 4.5) correspond à la transition inverse du cas précédent : le contexte de la nouvelle tâche de l'opérateur inclut celui de la tâche initiale, il doit intégrer un ensemble d'éléments de situation plus important, englobant les éléments nécessaires à la réalisation de la tâche initiale.

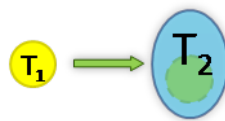


FIGURE 4.5 : transition simple – Extension de contexte.

Redondance totale des contextes

Ce cas particulier de transition (figure 4.6) traduit une correspondance exacte entre les éléments de situation nécessaire à la réalisation des deux tâches. Ce cas de figure peut se présenter lorsque T_1 et T_2 représentent une même et unique tâche itérée plusieurs fois. Lorsque la tâche se réinitialise, nous obtenons ce type de transition.

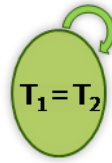


FIGURE 4.6 : transition simple – Redondance totale des contextes.

4.1.3.2 Ajout d'une tâche supplémentaire (type $T_1 \Rightarrow T_1 + T_2$)

Les 5 transitions suivantes appartiennent à la catégorie des « ajouts de tâche » : une nouvelle tâche T_2 , ayant un but différent de la première, est ajoutée et exécutée simultanément à la tâche T_1 .

Séparation des contextes

Dans ce cas de figure, la nouvelle tâche ajoute un nouvel ensemble d'éléments de situations, associés à la nouvelle tâche à exécuter, qui n'ont rien en commun avec la tâche initiale T_1 (figure 4.7).

Ce type de transition peut être très exigeant pour l'opérateur, les deux tâches à exécuter simultanément étant complètement différentes et font appel à des contextes différents, imposant de s'adapter à ces deux tâches.

Par exemple, dans le cas d'une conduite de centrale nucléaire, la supervision et la réalisation de tâches spécifiques en salle de commande (montée en puissance du réacteur par exemple) viennent s'ajouter des tâches « d'essais périodiques » nécessitant l'attention de l'opérateur sur des aspects du contrôle/commande entièrement différents de ceux de la tâche initiale.

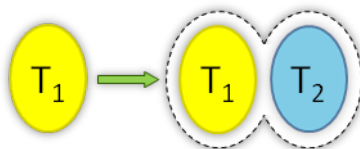


FIGURE 4.7 : ajout de tâche – Séparation des contextes

Intersection des contextes

Peut-être plus classique, cette transition (figure 4.8) consiste à ajouter une tâche partageant un certain nombre d'éléments de situation, pertinents à sa réalisation, avec la tâche initiale.

Par exemple, un régulateur de bus peut se voir attribuer la charge d'une ligne supplémentaire afin de décharger un collègue de travail (ayant par exemple besoin de se focaliser sur une autre ligne). Cette nouvelle ligne ajoute de nouveaux éléments contextuels mais certains éléments – un tronçon commun aux deux lignes – sont partagés avec l'activité initiale de l'opérateur.

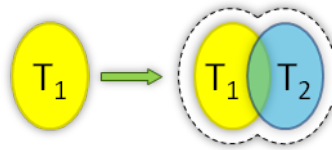


FIGURE 4.8 : ajout de tâche – Intersection des contextes

Réduction de contexte

Comme nous l'avons vu dans la section portant sur les « transitions simples », ce type de transition correspond à un focus de l'opérateur sur une tâche plus précise de l'activité. Ici, à la différence des transitions simples, la tâche « plus précise » est exécutée en parallèle avec la tâche initiale, aucun élément de situation supplémentaire n'est apporté par cette transition (figure 4.9).

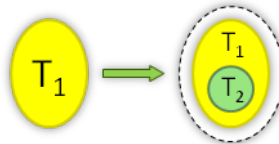


FIGURE 4.9 : ajout de tâche – Réduction de contexte

Extension de contexte

Transition inverse de la précédente, cet ajout de tâche apporte de nouveaux éléments de situation a priori connectés avec ceux nécessaires à la réalisation de la tâche T_1 , élargissant le champ contextuel de cette tâche avec l'exécution d'une tâche plus « globale ».

Cette transition (figure 4.10) pourrait correspondre à l'exemple d'un superviseur contrôlant un essaim de drones et devant reprendre une tâche de supervision globale tout en gardant un œil sur l'activité d'un drone en particulier (qui est proche d'une zone à risque par exemple).

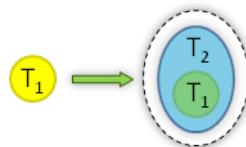


FIGURE 4.10 : ajout de tâche – Extension de contexte

Redondance totale des contextes

Enfin, cette transition correspond au cas où la nouvelle tâche partage entièrement ses exigences en termes de *Situation Awareness* avec la tâche initiale.

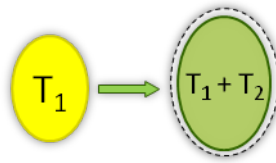


FIGURE 4.11 : ajout de tâche – Redondance totale des contextes

4.1.3.3 Le cas des interruptions (type $T_1 \Rightarrow T_2 \Rightarrow T_1$)

Comme nous l'avons énoncé, ces transitions correspondent à un enchaînement de deux transitions simples symétriques : la tâche T_2 interrompt T_1 avant de s'effacer pour laisser T_1 reprendre. Nous présentons dans cette sous-section les 5 combinaisons de transitions simples permettant d'obtenir des interruptions

Séparation des contextes

Cette interruption est composée de deux transitions simples inverses où les contextes associés à T_1 et T_2 sont disjoints (figure 4.12). Cette disjonction traduit le fait qu'un opérateur voit sa tâche principale interrompue par une tâche secondaire n'ayant rien en commun.

Un exemple de ce type d'interruption pourrait être celui d'un commandant de bord d'aéronef civil, lors d'une phase de croisière, déléguant la supervision du vol à son copilote afin de remplir des tâches administratives sur un autre système, puis se concentrant à nouveau sur le vol, et doit donc reconstruire sa SA sur les paramètres de l'avion.

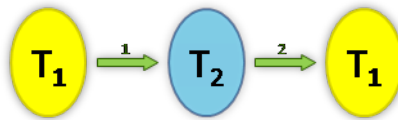


FIGURE 4.12 : interruption de tâche – Séparation des contextes

Intersection des contextes

Ce type d'interruption est déclenché par des tâches T_2 dont le contexte informationnel est proche de celui la tâche initiale.

Il peut s'agir par exemple de tâche de type « gestion d'erreur » ou « d'alarme », nécessitant de la part de l'opérateur une attention totale et immédiate, sur des aspects de la mission proches de la situation associée à la tâche initiale, certaines parties de ce contexte initial pouvant aider l'opérateur à réaliser la tâche secondaire.

Réduction de contexte

Cette interruption est la combinaison d'une transition simple de type « réduction »

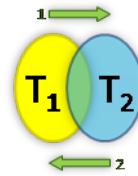


FIGURE 4.13 : interruption de tâche – Intersection des contextes

et d'une transition simple de type « extension » (dans cet ordre, voir figure 4.14). Nous avons fait apparaître en transparence les éléments de situation associés à T_1 lors de l'interruption par T_2 car ceux-ci, à la différence d'une transition simple, vont être nécessaires à la reprise de T_1 . De la même façon, nous faisons apparaître les éléments de T_2 dans la deuxième occurrence de T_1 , symbolisant le fait que ces éléments sont partagés par les deux tâches lors de la reprise et peuvent servir de « point d'ancrage » pour que l'opérateur ré-intègre l'ensemble des éléments de situation associés à la tâche initiale.

Par exemple, dans le cas de la supervision de centrale nucléaire, l'occurrence d'un fortuit (panne) sur un équipement particulier peut nécessiter l'attention totale de l'opérateur, au détriment de la tâche initiale de supervision, lui imposant de reconstruire sa SA au retour sur la tâche initiale, notamment sur les éléments ne touchant pas la panne.

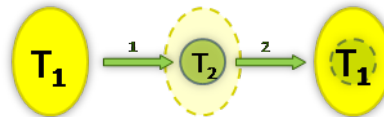


FIGURE 4.14 : interruption de tâche – Réduction de contexte

Extension de contexte

Cette interruption est la combinaison d'une transition simple de type « extension » et d'une transition simple de type « réduction » (dans cet ordre, voir figure 4.15). De la même façon que pour l'interruption précédente, nous avons fait apparaître les éléments de situation nécessaires à la reprise en mémoire. Cette interruption consiste pour l'opérateur à réaliser une tâche secondaire ayant un contexte plus large que sa tâche initiale mais s'appuyant sur un noyau d'éléments de situation associé à cette tâche initiale. Ainsi, tout au long de l'interruption, les éléments associés à T_1 font partie des exigences en termes de SA.

Redondance totale des contextes

Lorsque T_1 est interrompue par T_2 , les éléments de situation nécessaires à l'exécution de cette nouvelle tâche secondaire sont exactement les mêmes que ceux nécessaires à la réalisation de la tâche initiale.

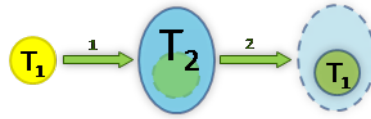


FIGURE 4.15 : interruption de tâche – Extension de contexte

Ce type d'interruption peut par exemple représenter une interruption de la tâche principale par un outil de mesure de type SAGAT (§ 3.4.2.1) permettant de mesurer la SA : l'ensemble des éléments de situation de la tâche principale sont nécessaires à la réalisation de la tâche secondaire (le but de cette tâche étant d'évaluer la connaissance de ces éléments par l'opérateur). Une telle tâche peut être intégrée à l'étude de conception de systèmes tels que des simulateurs de pilotage, car elle fait partie des tâches décrites dans l'activité de l'opérateur de tels systèmes.

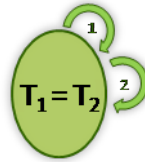


FIGURE 4.16 : interruption de tâche – Redondance totale des contextes

4.1.3.4 Tableau de synthèse

Nous avons proposé dans cette section une taxinomie des transitions permettant d'approcher de manière qualitative l'analyse des transitions pouvant se produire dans un système complexe. Le tableau 4.2 présente les 15 types de transitions présentées dans cette section, obtenues en parcourant les différentes classes des deux axes temporel et contextuel.

Cette taxinomie peut être utilisée afin de catégoriser les différentes transitions identifiées lors de la phase d'analyse et de description de l'activité des opérateurs pour le système considéré. Le rattachement des transitions à ces différentes classes pourrait par exemple être utilisé pour prédire l'impact qu'auraient ces transitions sur l'opérateur : augmentation de la charge cognitive due à l'ajout d'informations à gérer, perte de focus sur un contexte général, interférence des informations liées à d'anciennes tâches n'étant plus nécessaires pour la réalisation de la tâche courante, erreurs dues à la proximité des contextes, etc. Nous développons cette notion dans le paragraphe 4.3.1.3, appliquée au modèle dual présenté dans la section suivante.

4.2. VERS UN NOUVEL ESPACE DE MODÉLISATION DES TRANSITIONS

TABLEAU 4.2 : synthèse des transitions décrites sur les axes temporel et contextuel.

↓ Contexte Temps →	Transition simple	Ajout de tâche	Interruption
Séparation			
Intersection			
Réduction			
Extension			
Redondance totale			

4.2 VERS UN NOUVEL ESPACE DE MODÉLISATION DES TRANSITIONS

Afin de concrétiser l'approche duale adoptée pour l'analyse des transitions, nous proposons dans cette section de définir un nouveau modèle, reliant modélisation des tâches et de la *Situation Awareness*, sur lequel pourra s'appuyer l'expert en charge de l'analyse pour évaluer les éléments de situation associés aux transitions. Ce nouveau modèle s'inspire du modèle de *Distributed Situation Awareness* (DSA) défini par Stanton *et al.* (2006).

4.2.1 Préambule : le modèle DSA

Avant de présenter notre modèle permettant la description des transitions, nous rappelons les éléments clés du modèle DSA proposé par Stanton *et al.* (2006) (voir aussi § 3.3.2).

Le modèle DSA permet de modéliser la SA distribuée entre différents opérateurs d'un système collaboratif. Il s'appuie sur la modélisation, à travers trois réseaux, des relations entre les opérateurs (réseau social), des relations entre les tâches et leur attribution à chaque opérateur (réseau de tâche) et des relations entre les objets de connaissances, représentant la SA associée à chaque tâche, décrits sous forme de réseau propositionnel (réseau de connaissance).

Chaque tâche du réseau de tâches est attribuée à un ou plusieurs opérateurs selon des rôles pré-définis. Un sous-ensemble d'objets de connaissance du réseau de connaissances est également associé à l'exécution de chacune des tâches (quelque soit le nombre d'opérateurs réalisant la tâche). Cet ensemble définit les concepts contribuant à la construction de la SA pour la tâche (décrivant ainsi ce que nous avons introduit sous le terme de « contexte » au début de ce manuscrit). Cette association est visualisée à travers la coloration d'éléments du réseau propositionnel en fonction des tâches étudiées (dans (Stanton *et al.*, 2006) une couleur est associée à chaque tâche). La figure 4.17 illustre cette association entre deux tâches du réseau (attribuées généralement à deux opérateurs distincts, non représentés sur la figure) et les objets de connaissances nécessaires pour la réalisation de chacune de ces tâches.

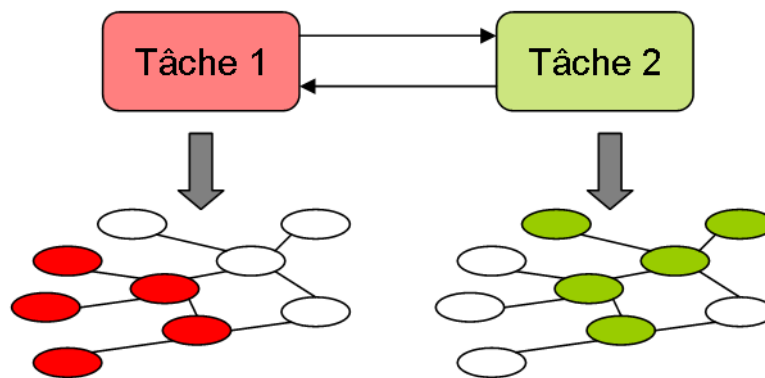


FIGURE 4.17 : DSA : exemple d'association entre deux tâches du réseau de tâche et les objets de connaissance du graphe propositionnel.

4.2.2 Critiques et adaptation du modèle DSA

4.2.2.1 Points forts du modèle

Dans sa modélisation de la SA des opérateurs d'un système collaboratif, le modèle DSA présente plusieurs avantages par rapport à d'autres modèles plutôt orientés processus, basé sur les concepts de SA d'équipe et de SA partagée.

La modélisation des 3 niveaux d'étude (social, tâches et connaissance) à travers 3 réseaux permet de concrétiser de manière formelle les différents concepts mis en jeu dans l'analyse de la SA de tels systèmes. A la différence de travaux comme ceux de Saner *et al.* (2009) où les exigences de SA sont définies comme des vecteurs d'éléments indépendants, décrits sous forme textuelle, le réseau de connaissances du modèle DSA permet de visualiser de manière globale l'ensemble des éléments

de situation exigés pour un bon fonctionnement du système dans sa globalité et les tâches prises en charge par chaque opérateur, tout en décrivant de manière visuelle les liens entre ces objets de connaissance, adoptant une approche globale et distribuée du concept de SA (quand l'approche par vecteur ne permet de visualiser les éléments associés à chaque opérateur que sous forme de tableaux, sans liens entre les différents éléments).

Dans le modèle DSA, le principe d'association entre objets de connaissance du réseau propositionnel et les tâches considérées, visualisé à travers la coloration des nœuds du réseau, contribue également à faciliter la description et la compréhension des exigences pour chaque opérateur/tâche. De plus, cette représentation sous forme de réseau répond à des justifications théoriques, fondant la théorie de la DSA sur la représentation de schémas mentaux, dont la construction satisfait une logique de résolution de problèmes.

Enfin, nous notons que l'aspect visuel et l'ancrage théorique du modèle DSA sont proches des idées sous-tendant le concept de *mindmapping* (carte heuristique), utilisé par les ergonomes lors des phases d'analyse de l'activité afin de représenter les cheminements de pensée des opérateurs.

4.2.2.2 Limites du modèle

Dans le modèle DSA, l'ensemble des tâches réalisées par les opérateurs du système est modélisé sous forme d'un réseau. Ce réseau se traduit par un ensemble de blocs, reliés entre eux par des flèches uni- ou bi-directionnelles selon les enchaînements valides pour l'activité considérée. Peu de détails supplémentaires sont précisés sur ce modèle de tâches, qui n'est pas décrit formellement dans les différents travaux des auteurs, contrairement au modèle de connaissance. Ainsi, dans cette représentation des tâches en réseau, les transitions entre les différentes tâches sont uniquement visualisées par les flèches et aucune précision n'est apportée par le modèle sur le déroulement temporel de l'activité.

De plus, comme ce réseau de tâches est commun à l'ensemble des opérateurs du système, il faut pouvoir identifier quelles sont les tâches attribuées à chaque rôle. Dans les différents cas d'application disponibles à travers les articles relevés, cette attribution des tâches à des rôles semble être bijective : le niveau d'abstraction élevé des tâches permet de faire correspondre chacune d'entre elle à un rôle-opérateur ou de créer des sous-équipes par groupes de tâche, l'attribution des rôles étant alors visualisée dans un tableau (Stanton *et al.*, 2009, tableau 1).

Nous notons donc que l'intégration de la modélisation des tâches dans la théorie DSA présente des limites vis-à-vis de notre besoin de modélisation précise des tâches opérateurs et, en particulier, celle des transitions entre ces tâches. Il faut également

rappeler que, dans le cadre de cette thèse, nous nous focalisons sur les modèles *mono-opérateurs* et *multi-tâches*, tandis que la DSA a été initialement conçue pour répondre à des problématiques de systèmes *multi-opérateurs*, chaque opérateur ayant un rôle bien défini dans le système (auquel on associe des éléments de situation).

4.2.2.3 Adaptation du modèle DSA

Nous proposons ainsi d'adapter le modèle DSA à nos besoins afin de permettre l'analyse des transitions à travers l'approche duale tâche / contexte.

La première limite identifiée porte sur l'aspect multi-opérateur du modèle initial, qui ne correspond pas à notre focus sur des systèmes mono-opérateur. Le paradigme multi-opérateur / mono-tâche du modèle DSA permet de décrire la multitude de tâches prises en charge par l'équipe d'opérateurs, le lien entre ces tâches et les éléments de situation associés à chacune d'entre elles : chaque opérateur adopte un point de vue différent sur la situation globale, afin de réaliser sa (ou ses) tâche(s).

Nous transposons ce paradigme au cas mono-opérateur / multitâche : un unique opérateur adopte plusieurs points de vue sur son activité, en fonction des contextes définis pour chaque tâche (figure 4.18). Ainsi, pour un modèle de tâches décrivant l'activité d'un opérateur unique, un contexte, défini à partir d'un sous-ensemble de nœuds du réseau propositionnel global, est associé à chacune des tâches du modèle, traduisant les exigences en termes de *Situation Awareness* pour la tâche considérée.

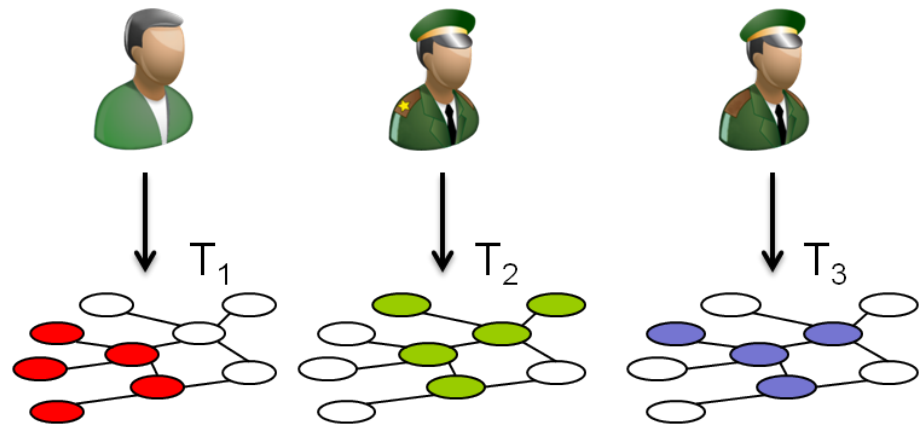
Cette transposition à un paradigme mono-opérateur nous permet également de mettre de côté le réseau social représentant les relations entre les opérateurs du système collaboratif, ce réseau n'étant plus pertinent dans notre contexte.

Le modèle de tâches proposé par Stanton et ses collègues dans leur approche initiale présente néanmoins également quelques limitations, notamment dans la description des transitions entre les différentes tâches, et a fortiori si nous devons décrire l'ensemble de l'activité et l'enchaînement des tâches d'un opérateur unique. Nous proposons de formaliser la représentation des tâches à travers l'utilisation d'une notation, sélectionnée parmi celles décrivant les modèles existants (dont nous avons présenté une sélection dans la section 2.1).

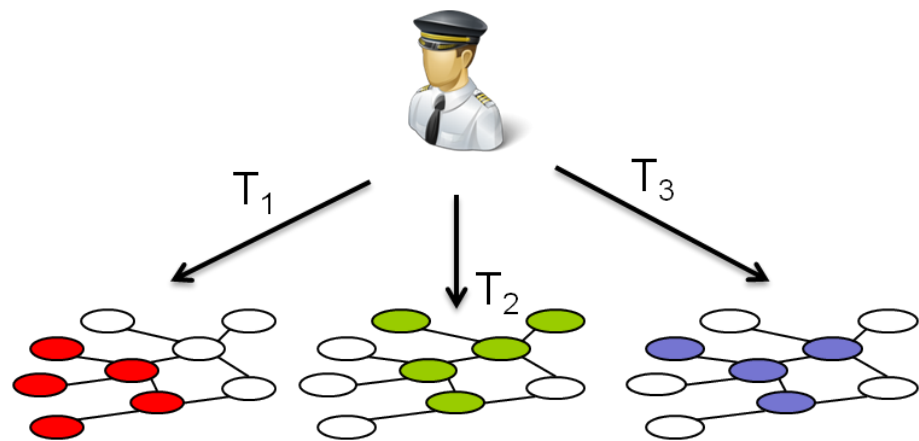
Dans la section suivante, nous présentons le choix du modèle de tâches et de la notation associée sur lesquels nous nous reposerons pour compléter la représentation des transitions.

4.2.3 Choix du modèle de tâches

Dans la section 2.1, synthétisée dans le tableau 2.1, nous avons présenté un ensemble de notations, textuelles ou graphiques, permettant de modéliser les tâches



(a) dans le modèle DSA, chaque opérateur adopte un point de vue différent sur la situation, dépendant de sa tâche.



(b) dans notre modèle, un opérateur unique adopte plusieurs points de vue sur la situation, en fonction de la tâche qu'il exécute.

FIGURE 4.18 : la transposition du paradigme multi-opérateur / mono-tâche vers le paradigme mono-opérateur / multitâche. En fonction des tâches exécutées par l'opérateur unique, celui-ci doit construire une SA basée sur les éléments des contextes associés.

en adoptant une approche centrée sur l'utilisateur. Afin de sélectionner la notation qui nous paraît la plus adaptée à l'analyse des transitions, nous avons établi plusieurs critères :

Décomposition hiérarchique des tâches

Le niveau de description des tâches adopté dans nos travaux correspond à celui utilisé dans les réseaux de tâche du modèle DSA, les tâches sont décrites de manière globale et, pour l'analyse des transitions entre ces tâches, nous pouvons nous restreindre à un niveau d'abstraction élevé.

Néanmoins, dans le cadre de l'ingénierie dirigée par les modèles (IDM), il peut être intéressant de faciliter la description plus concrète des tâches considérées,

afin notamment d'intégrer les résultats de l'analyse des transitions sous forme de tâche « atomiques » décrivant les interactions entre l'utilisateur et le système.

Afin de répondre à ces deux besoins, la notation sélectionnée doit donc permettre la décomposition des tâches de manière hiérarchique avec différents niveaux d'abstraction.

Intégration des relations temporelles entre tâches

Afin de faciliter l'analyse des transitions entre tâches, cœur de ce chapitre et objectif de notre thèse, il nous semble requis de choisir une notation intégrant ces relations. L'ensemble des notations présentées dans l'état de l'art (section 2.1) intègre ces relations temporelles entre les tâches. Néanmoins, pour une partie de ces notations, ces relations ne sont pas formalisées, comme dans GTA (van der Veer *et al.*, 1996) et/ou se limitent à la description de plans, comme pour HTA (Annett et Duncan, 1967) par exemple.

La formalisation des relations temporelles offre à l'expert en charge de la modélisation un choix fini et pré-défini de relations et assure l'homogénéité de la description de l'enchaînement des tâches, au sein d'un même modèle mais également à travers toute l'activité de modélisation d'un grand projet par exemple. Ainsi, nous éliminons les notations laissant à l'expert le soin de la définition des relations.

Nous favorisons également les notations intégrant des relations avancées, telle que l'interruption de tâche, dans la modélisation. Ce type de transition fait partie de la taxinomie définie en section 4.1 et il nous semble logique de l'intégrer dans la modélisation des tâches de l'activité de l'opérateur.

Pouvoir descriptif de la notation

Afin de faciliter les échanges avec les utilisateurs autour du modèle de tâches conçu (dans un processus de conception centré sur les utilisateurs), les notations représentées sous forme graphiques possèdent un pouvoir de description plus grand que les représentations textuelles.

Les notations graphiques présentées possèdent également l'avantage d'être appuyées par des éditeurs, facilitant la création et la modification des modèles par l'expert ayant cette charge.

Le tableau 4.3 reprend l'ensemble des 9 notations présentées au chapitre 2 et identifient celles qui répondent aux critères fixés ci-dessus.

De ce tableau, il ressort que seules deux notations correspondent à l'ensemble des critères définis : K-MAD (Lucquiaud, 2005b) et CTT (Paternò *et al.*, 1997), l'éditeur associé à la notation TOOD (Abed *et al.*, 2003) n'étant plus maintenu.

Néanmoins, dans le cas de l'analyse des transitions, nous remarquons que ces deux notations n'abordent pas la représentation des relations temporelles entre tâches de la même façon :

- CTT décrit directement la relation temporelle au niveau des tâches concernées : les deux tâches « sœurs » sont reliées entre elles par un opérateur *gra-*

4.2. VERS UN NOUVEL ESPACE DE MODÉLISATION DES TRANSITIONS

TABEAU 4.3 : comparaison de neuf notations vis-à-vis des critères de sélection définis.

Notation	HTA	GOMS	GTA	UAN	TKS	Diane+	CTT	K-MAD	AMBOSS	TOOD
Décomposition hiérarchique	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Représentation	Textuelle, graphique	Textuelle	Graphique	Textuelle	Textuelle	Graphique	Graphique	Graphique	Graphique	Textuelle, graphique
Éditeur associé	✗	✗	✓	✗	✗	✓	✓	✓	✓	✗
Relations formelles	✗	✓	✗	✓	✗	✓	✓	✓	✓	✓
Interruption	Non formel	✗	Non formel	Opérateur	Non formel	✗	Opérateur	Booléen	✗	Fonction

phique, issu de la notation LOTOS (ISO, 1989), permettant pour un même niveau hiérarchique de décrire l'enchaînement des tâches en intégrant plusieurs relations avancées différentes, parmi les 8 opérateurs proposés dans CTT. Notons que l'interruption d'une tâche par une autre est représentée par un opérateur dans la notation CTT, liant la tâche interrompue avec la tâche secondaire.

- K-MAD décrit la relation temporelle liant plusieurs tâches au niveau de leur tâche « mère ». Ainsi, une seule relation, décrite sous forme *textuelle* au niveau du nœud parent, permet de décrire la relation liant toutes les sous-tâches de ce nœud. Les quatre relations de base (alternative, parallélisme, séquence et ordre indéfini) sont proposées ainsi que la possibilité de définir la tâche comme « élémentaire » (tâche atomique). De plus, la possibilité d'interrompre une tâche n'est mentionnée que sous la forme d'un booléen, ne précisant pas quelles tâches peuvent interrompre la tâche considérée.

Ainsi, afin de soutenir la visualisation et l'analyse des transitions entre tâches, nous pensons que CTT offre davantage de possibilités que la notation K-MAD. En effet, sa structure d'arbre n-aire associée à la description des transitions entre les tâches d'un même niveau hiérarchique permet de décrire de manière visuelle l'enchaînement logique et temporel des tâches. Outre l'aspect visuel, permettant notamment de décrire précisément par quelle(s) tâche(s) peu(ven)t être interrompue une tâche spécifique, l'usage de la notation CTT est répandu, et notamment très utilisé au sein de l'équipe d'accueil du doctorant (par exemple, dans le cadre du projet présenté en annexe D (Leal *et al.*, 2009)). L'outil CTTE (Mori *et al.*, 2002), toujours maintenu et mis à jour régulièrement, permet d'assurer la diffusion et l'utilisation de la notation.

De plus, plusieurs autres notations reposent sur le formalisme de CTT et proposent des variantes afin de répondre à différentes problématiques de modélisation. En particulier, la notation COMM (Jourde *et al.*, 2010) nous semble particulièrement intéressante, notamment grâce au fait qu'elle intègre l'usage des relations de Allen en supplément des relations initialement proposées dans CTT, afin de compléter la description de l'enchaînement de tâches. La notation COMM est également appuyée par un éditeur, fonctionnant en ligne, appelé e-COMM (Jourde et Laurillau, 2010).

Enfin, la construction des arbres de CTT est formalisée par des règles de construction, portant notamment sur la validité des enchaînements temporels possibles, l'héritage des types de tâches ou encore la priorité entre les différents opérateurs temporels (Paternò *et al.*, 1997; Paternò, 2003).

4.2.4 Synthèse de l'approche duale

Afin d'appuyer la description des transitions dans la continuité de l'approche duale tâche / contexte présentée en introduction de ce chapitre, nous proposons d'adapter le modèle de *Distributed Situation Awareness* (Stanton *et al.*, 2006) afin de décrire l'ensemble des tâches de l'opérateur de système complexe et les exigences en terme de *Situation Awareness* associées à chaque tâche de l'opérateur.

Le modèle proposé repose sur l'association de deux modèles, déjà existants et la description de l'activité à travers ces deux modèles :

- Afin de décrire les tâches de l'opérateur, nous nous reposons sur la notation ConcurTaskTree (CTT, Paternò *et al.*, 1997) présentée dans les paragraphes 2.1.2.7 et 4.2.3.

Le modèle de tâches permet de décrire l'ensemble des tâches exécutées par l'opérateur, en respectant une décomposition hiérarchique (sur un axe vertical) et temporelle (sur un axe horizontal) des tâches et sous-tâches de l'activité. Afin de réaliser l'analyse des transitions, nous restreignons cette description aux niveaux d'abstraction les plus élevés, laissant de côté la description précise des interactions à une phase ultérieure, dépendant des résultats de l'analyse et correspondant à des solutions concrètes.

Nous décrivons en détail la construction du modèle de tâches à partir de la notation CTT dans la section 5.2.2 du chapitre suivant.

- Les exigences en terme de *Situation Awareness* sont décrites à partir du modèle DSA (Stanton *et al.*, 2006). L'ensemble des éléments de situation requis pour travailler sur le système sont représentés sous la forme d'un réseau propositionnel. La construction de ce réseau est décrite dans la section 5.2.3 du chapitre suivant.

A chaque tâche de l'arbre de tâche est associé un sous-ensemble de ce réseau propositionnel correspondant aux éléments de situations requis pour l'exécution de la tâche considérée. Cette association entre tâches et éléments de situation est traduite visuellement par la coloration des nœuds du réseaux propositionnel associés à la tâche étudiée. La figure 4.19 représente sous forme schématisée cette association : les tâches 1.1 et 1.2 de l'arbre de tâches sont deux tâches séquentielles (enchaînement représenté par l'opérateur \gg), à chacune est associé un sous-ensemble d'éléments de situation du réseau propositionnel. Les éléments associés à la tâche 1.1 sont colorés en rouge tandis que ceux associés à la tâche 1.2 sont colorés en vert.

La description de l'ensemble de l'activité de l'opérateur se fait à travers la

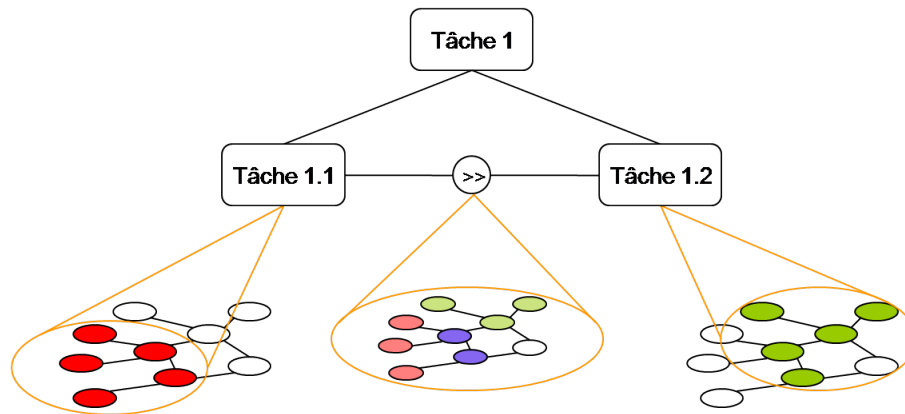


FIGURE 4.19 : principe d'association entre les éléments de situations du réseau propositionnel et les tâches de l'arbre CTT.

construction d'un arbre de tâche complet, à partir duquel on pourra décrire des scénarios types rencontrés par l'opérateur lors du déroulement de sa mission (§ 4.3.1.1). Pour chaque tâche de l'arbre, un ensemble d'exigences en terme de SA est défini et associé aux tâches.

En étudiant les éléments associés aux deux tâches d'une transition, il est possible de construire pour chaque transition un réseau propositionnel associé à cette transition (représenté au centre de la figure 4.19), permettant d'accéder aux éléments communs aux deux tâches (éléments colorés en mauve), aux éléments non-exigés pour la nouvelle tâche (rouge transparent) et aux nouvelles exigences de SA (vert transparent).

Dans la suite de ce chapitre, nous présentons plusieurs usages possibles pour l'analyse des transitions s'appuyant sur la taxinomie définie dans la section 4.1 et l'espace de modélisation dual que nous venons de définir dans la section courante.

4.3 DEUX USAGES POUR L'ANALYSE DES TRANSITIONS : PRÉDICTION ET AIDE À LA CONCEPTION

Les deux sections précédentes nous ont permis de définir une taxinomie des transitions et de proposer un espace pour modéliser les tâches et contextes associés à ces tâches. A présent, nous abordons le cœur de la problématique de thèse, à savoir l'analyse des transitions dans les systèmes complexes. Afin d'étayer cette problématique, nous avons l'avons divisée en deux questions reformulées ici :

- Comment peut-on prévoir les potentielles ruptures de *Situation Awareness* induites par les changements de tâche et mettre en exergue celles qui pourraient avoir un impact sur l'opérateur ?
- Quels sont les éléments à prendre en compte afin d'améliorer l'interface conçue

et accompagner au mieux l'opérateur lors de ces transitions ?

Ainsi, à travers ces deux problématiques nous identifions deux approches :

- une approche **prédictive**, basée sur l'analyse globale des scénarios et l'identification des transitions difficiles ;
- une approche d'**aide à la conception**, visant à approfondir l'analyse locale de certaines transitions afin de définir quels sont les éléments pouvant impacter la *Situation Awareness* de l'opérateur et proposer des solutions d'intégrations de ces éléments dans la conception de l'interface.

Dans la suite de cette section (§ 4.3.1), nous présentons des pistes pour la *prédiction* des transitions difficiles. Nous consacrerons le chapitre suivant à l'approche « locale » et proposerons notamment une méthodologie d'analyse des transitions pour la conception d'interface.

4.3.1 Réflexions sur l'analyse prédictive des scénarios

4

La taxinomie des transitions introduite dans la section 4.1 identifie 15 types de transitions « atomiques », caractérisées par deux dimensions : temporelle et contextuelle. Chaque transition possède ainsi ses propres caractéristiques et leur impact sur la construction de la *Situation Awareness* des opérateurs est différente. Nous proposons dans cette section quelques pistes de réflexion sur la manière dont un expert, en charge de l'analyse de l'activité de l'opérateur et de la conception du système futur, pourrait s'appuyer sur cette taxinomie et l'approche duale tâche / contexte afin d'identifier les transitions pouvant induire un cas de rupture de la SA (transitions que nous qualifierons de « difficiles » dans la suite de ce chapitre).

Cette démarche d'analyse prédictive est appuyée par la définition de scénarios types que nous définissons en préambule de cette section. Ceux-ci ont pour but de focaliser l'analyse de l'expert sur les transitions pertinentes dans le cadre opérationnel. Nous proposons ensuite deux pistes de réflexion pour l'analyse « globale » des scénarios définis :

- une **analyse qualitative**, s'appuyant sur la taxinomie et l'estimation de la charge cognitive induite par les différentes transitions ;
- une **analyse quantitative**, basée sur le calcul d'un score de similarité entre tâches afin de quantifier l'impact éventuel des transitions et permettre de les classer par ordre de difficulté attendu.

4.3.1.1 Préambule : construction de scénarios

La complexité des systèmes considérés dans nos travaux conduit à la définition d'un nombre important de tâches lors de la construction du modèle de tâches associé. Nous avons proposé dans la section précédente de baser cette modélisation sur

la notation ConcurTaskTrees, afin de décrire l'ensemble des tâches en jeu¹. Mais l'analyse de l'ensemble des transitions de l'arbre des tâches représente un travail coûteux en temps, dont la difficulté augmente proportionnellement au nombre de tâches décrites.

Qui plus est, comme nous l'avons indiqué en introduction de ce mémoire, l'opérateur d'un système complexe est en charge d'un *ensemble* de tâches évoluant dynamiquement dans le temps. Ainsi, afin de simplifier l'analyse des transitions, nous adoptons une approche pragmatique basée sur la définition de scénarios types, permettant de prendre en compte dans l'analyse uniquement les transitions opérationnellement plausibles. Ces transitions sont définies à partir de l'analyse de l'activité et des besoins conduite en phase amont.

Ces scénarios décrivent la suite de transitions « atomiques » (définies dans la taxinomie) qui permet de passer d'un ensemble de tâches $I = \{T_{1..i}\}$ à un ensemble de tâches $J = \{T_{j..n}\}$ avec $(i, j) \in [1, n]^2$. En effet, plusieurs suites de transitions sont possibles pour décrire le passage d'une configuration à une autre, mais toutes ne sont pas pertinentes. Nous proposons donc de décomposer ces « méta-transitions » en suites de transitions atomiques, correspondant aux transitions de la taxinomie. Le but de cette décomposition est de clarifier le déroulement de l'activité et l'enchaînement prescrit des tâches afin de permettre d'en analyser les transitions.

Nous illustrons ce concept à travers la figure 4.20 où la tâche 3 (T_3) est exécutée en parallèle des tâches 1 (T_1) et 2 (T_2).

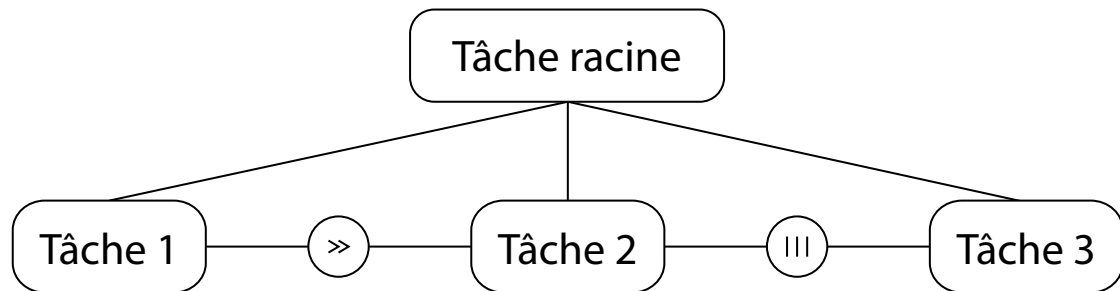


FIGURE 4.20 : exemple d'arbre de tâche décrivant 2 tâches séquentielles exécutées en parallèle d'une troisième tâche.

Ainsi, si la tâche 1 s'exécute seule en début de scénario, et qu'en fin de scénario, les tâches 2 et 3 sont exécutées en parallèles, il y a deux chemins possibles pour atteindre cette configuration finale :

- Soit T_2 est activée, désactivant T_1 puis T_3 est ajoutée dans un deuxième temps. On a alors les 2 transitions suivantes qui s'enchaînent $T_1 \rightarrow T_2$ (transition simple) et $T_2 \rightarrow T_2 + T_3$ (ajout de tâche). L'ensemble des tâches évoluerait

1. La section 5.2 présente en détails la façon dont est construit ce modèle.

alors ainsi :

$$\{T_1\} \rightarrow \{T_2\} \rightarrow \{T_2, T_3\}$$

- Soit T_1 continue de s'exécuter, T_3 commence, puis T_2 s'active, désactivant T_1 . Le scénario correspond alors à l'enchaînement des deux transitions : $T_1 \rightarrow T_1 + T_3$ (ajout de tâche) suivie de $T_1 \rightarrow T_2$ (transition simple, avec T_3 toujours exécutée en parallèle). L'ensemble des tâches évoluerait alors ainsi :

$$\{T_1\} \rightarrow \{T_1, T_3\} \rightarrow \{T_2, T_3\}$$

A travers cet exemple, nous voulons illustrer la possibilité de décomposer les différentes « méta-transitions » entre *ensembles* de tâches que les opérateurs ont à leur charge à tout instant de la mission. Nous remarquons que le principe de décomposition d'une « méta-transition » en une série de transitions « atomiques » peut se faire indépendamment de toute modélisation formelle des tâches. De plus, notons que nous n'avons ici illustré que l'aspect « temporel » de la décomposition des tâches mais l'aspect contextuel devrait également être pris en compte dans la définition des scénarios.

Afin de lever les ambiguïtés et limiter le nombre de possibilités de transitions, nous pensons qu'il est important de se reposer sur des scénarios décrivant à chaque instant les différentes tâches en jeu pour l'opérateur ainsi que les transitions à venir.

4.3.1.2 Introduction d'un exemple illustratif

Dans le but de concrétiser ce concept de scénario type, nous donnons un exemple succinct appliqué au cas du contrôle aérien militaire et plus particulièrement à la modélisation des tâches d'un contrôleur en charge de l'*approche* des avions de chasse rejoignant une base aérienne militaire. Ce contrôleur est en charge de la gestion d'une zone de vol militaire, activée auprès de la Direction Générale de l'Aviation Civile (DGAC), lorsque des vols militaires sont prévus.

Description des tâches

La figure 4.21 décrit l'arbre des tâches CTT définissant les tâches de l'opérateur. La section 5.2.2 présente une description détaillée de la construction et signification des arbres CTT.

Scénario type

Dans le cadre de son activité, le contrôleur aérien d'approche est amené à gérer un certain nombre d'avions militaires et à les guider jusqu'à un point situé environ 10 miles nautiques en amont de l'axe de la piste d'atterrissage, où il est transféré à un contrôleur responsable de la « finalisation » de l'approche et de l'atterrissage (dont le statut dépend de la méthode de guidage choisie par le pilote).

Un scénario type de guidage se déroule ainsi :

4.3. ANALYSE DES TRANSITIONS : PRÉDICTION ET AIDE À LA CONCEPTION

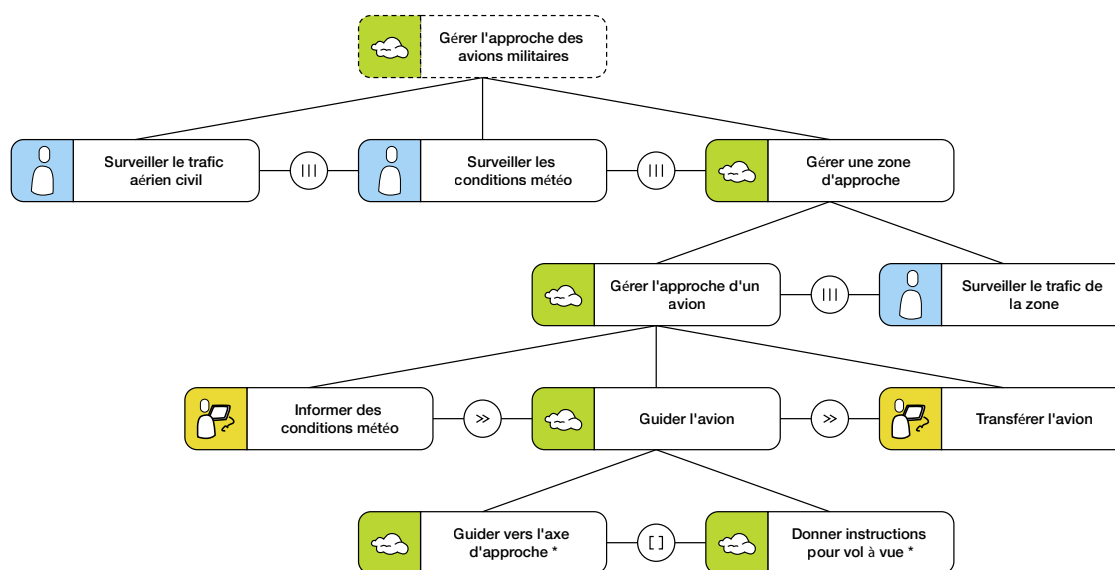


FIGURE 4.21 : description des tâches d'un opérateur d'approche dans le cadre de la circulation aérienne militaire.

1. Le contrôleur prend en charge un avion auquel il communique les conditions météorologiques de la base d'arrivée. Les conditions météo sont mauvaises (code « ambre »), la visibilité est mauvaise.
2. Le contrôleur doit alors guider le pilote dans l'axe de la piste où il devra atterrir en mode PAR (*Precision Approach Radar*), et être pris en charge par un opérateur différent.
 - (a) Le contrôleur, en s'appuyant sur les informations radars disponibles, portant sur le trafic civil et militaire aux alentours, donne une série de consignes de cap et d'altitude au pilote, afin qu'il se place dans l'axe de la piste.
 - (b) Il accompagne le pilote jusqu'à son arrivée au point de contact, 10 miles nautiques en amont de la piste, à 1 500 pieds d'altitude.
3. Il transfère alors l'avion au contrôleur aérien « responsable PAR », qui prend la main afin de finaliser l'approche et guider le pilote dans l'atterrissage.

Le scénario décrit ci-dessus permet d'illustrer les différents états dans lequel va se trouver le contrôleur aérien responsable de l'approche. Tout au long du scénario, certaines tâches sont présentes en tant que « tâches de fond » : le contrôleur doit à la fois gérer l'avion sur lequel il est focalisé, mais aussi prendre en compte la circulation aérienne militaire et civile afin d'anticiper tout risque potentiel.

Les différentes tâches du scénario, exécutées dans cet ordre, vont ainsi définir des transitions « atomiques » que l'on peut étudier et analyser plus en profondeur. La

figure 4.22 décrit l'enchaînement des tâches du scénario de guidage d'approche d'un avion militaire.

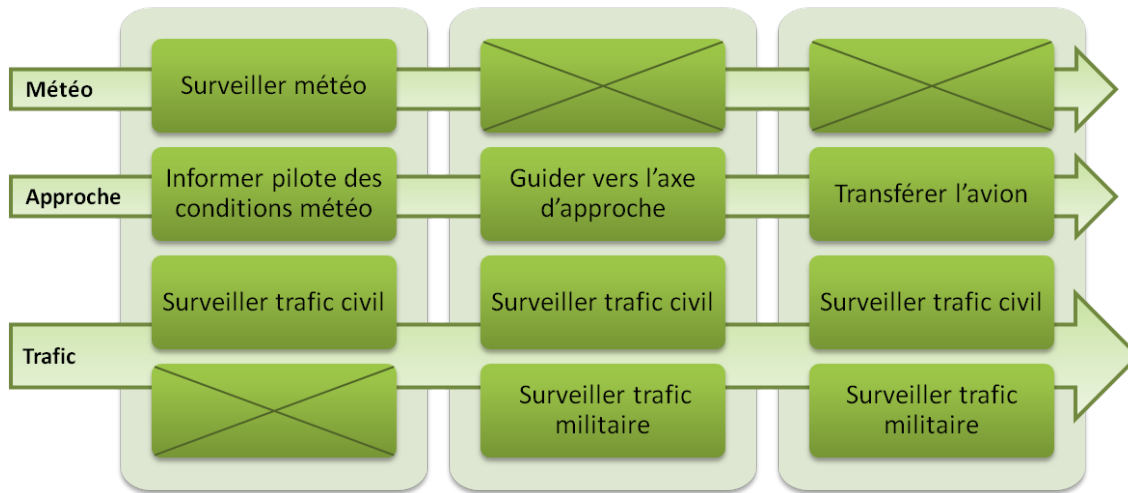


FIGURE 4.22 : description des tâches en jeu dans le scénario de contrôle aérien militaire.

Dans la suite de cette section, nous posons l'hypothèse que l'expert en charge de l'analyse travaille sur les scénarios types définis au préalable. Les réflexions proposées reposent sur le principe de « filtrage » amont des transitions non pertinentes de ces scénarios. Ce filtrage est basé sur l'analyse de leurs impacts sur la *Situation Awareness* de l'opérateur.

4.3.1.3 Application de la taxinomie au filtrage des transitions

Notre première réflexion propose une **analyse qualitative** des transitions des scénarios, basée sur la classification des transitions en fonction de leur difficulté pour l'opérateur. En effet, dans notre état de l'art portant sur le *Task Switching* (section 2.2) et les interruptions (section 2.3), nous avons noté que les transitions entre ces tâches « simples » (c'est-à-dire à un niveau d'abstraction faible) avaient un impact sur la performance de l'opérateur (dont la mesure diffère selon le paradigme expérimental). En extrapolant ces réflexions à notre cas de figure, où l'approche des transitions se focalise sur la construction de la *Situation Awareness* associée à chaque tâche, nous pouvons imaginer que l'impact sur la SA sera différent en fonction du type de transition étudié.

Ainsi, en identifiant les transitions intervenant dans les scénarios types définis avec les transitions caractéristiques de la taxinomie, nous pourrions définir les priorités d'étude des transitions du scénario compte tenue de la difficulté escomptée de chaque transition. En identifiant chacune des transitions du scénario à l'une des 15 transitions de la taxinomie, l'expert peut ainsi éliminer celles jugées *a priori* peu

problématiques afin de focaliser son analyse sur les transitions restantes.

Pour permettre d'appliquer cette approche, il faut néanmoins estimer la difficulté associée à chaque transition, métrique non définie, et potentiellement difficile à définir.

En nous basant sur les travaux présentés, nous pouvons supposer par exemple que les transitions de type *interruption* sont plus difficiles à que les *transitions simples*. De la même façon, l'ajout d'une tâche induit généralement une charge de travail supérieure. Nous pouvons également penser que les transitions de *réduction* du contexte (focalisation sur un sous-ensemble du graphe initial) n'impliquent pas de nouveaux éléments de situation, et sont peut-être moins difficiles que des transitions où il n'y a aucun élément en commun.

L'étude de l'impact de chaque type de transition est néanmoins un travail fastidieux et l'ébauche de réflexion présentée ici se devrait d'être étayée et validée par la littérature (voir par exemple Chanquoy *et al.*, 2007) et des expérimentations permettant de comparer les différents types de transitions et l'impact sur la SA d'un opérateur de l'exécution de deux tâches T_1 et T_2 selon différentes configuration de transitions.

L'usage de la taxinomie pour le filtrage des transitions et les réflexions à porter sur l'impact cognitif des différentes catégories de cette taxinomie constituent une étude à part entière que nous n'abordons pas dans le cadre de cette thèse. Ces pistes de réflexions pourra nourrir de futurs travaux permettant, en amont d'une analyse précise des transitions mises en jeu dans un scénario opérationnel type, de faire un tri parmi les transitions difficiles ou non, afin par exemple de définir de manière qualitative des priorités pour cette analyse.

4.3.1.4 Similarité des tâches et classement des transitions

Nous présentons dans ce paragraphe une deuxième approche de « filtrage » des transitions, avec toujours pour objectif de limiter le nombre de transitions à analyser précisément ensuite, permettant de se focaliser sur les transitions les plus pertinentes en termes d'impact sur la *Situation Awareness* de l'opérateur.

L'approche proposée consiste en une **analyse quantitative** des transitions, et des éléments de situation associés aux tâches constituant la transition. Le principe est, pour chaque transition « atomique » d'un scénario, de calculer un *score de similarité* entre les contextes associés aux deux tâches impliquées dans la transition considérée, puis d'ordonner les transitions ainsi caractérisées en fonction de ce score.

Score de similarité de Tversky

Dans ses travaux fondateurs sur les propriétés de la relation de similarité, Tversky (1977) introduit le *contrast model*, décrivant la « similarité entre différents objets

comme la différence pondérée de la mesure de leurs caractéristiques communes et distinctes ». Tversky applique son modèle à la catégorisation d'objets en fonction de leurs caractéristiques, définissant par exemple des familles de ressemblances pour ces objets et étudiant la prototypicalité de ces objets (les objets ayant toutes les caractéristiques propres à une classe). Il propose notamment des arbres hiérarchiques de caractéristiques dont les branches, au fur et à mesure de leur déploiement, précisent les caractéristiques des objets y figurant.

Dans le cadre de notre étude, nous utilisons un score normalisé de similarité basé sur le *contrast model* : le ratio de similarité de Tversky (1977). A travers ce score (noté S), nous allons comparer deux à deux les tâches des scénarios types considérés, mettant en relation les éléments de situation communs et distincts des contextes associés aux deux tâches composant une transition.

Dans la suite de ce paragraphe, T_i et T_j sont les deux tâches comparées, et f la fonction permettant de définir le nombre d'éléments partagés et propres à chaque tâche.

Le ratio de Tversky est défini par l'équation 4.1 :

$$S(T_i, T_j) = \frac{f(T_i \cap T_j)}{f(T_i \cap T_j) + \alpha f(T_i \setminus T_j) + \beta f(T_j \setminus T_i)} \quad (4.1)$$

Ce ratio présente les généralités suivantes :

- Sa valeur est comprise entre 0 et 1.
- La fonction f définit un score quantitatif, dénombrant le nombre d'éléments associés à l'intersection ou la différence des ensembles T_i et T_j .

Dans le cas de la comparaison des contextes des tâche T_i et T_j , la fonction f comptabilise, pour chaque tâche, le nombre d'éléments de situation associés (nombre de nœuds colorés dans le réseau propositionnel global) :

- $f(T_i \cap T_j)$ représente le nombre d'éléments de situation communs aux tâches T_i et T_j .
- $f(T_i \setminus T_j)$ représente le nombre d'éléments associés à T_i mais pas à T_j .
- $f(T_j \setminus T_i)$ représente le nombre d'éléments associés à T_j mais pas à T_i .

Enfin, les coefficients α et β sont positifs ou nuls. Ils sont introduits par Tversky (1977) pour permettre de rendre le score asymétrique dans le but de donner une importance particulière aux éléments propres à l'un ou l'autre des contextes (et non partagés par les deux tâches). Tversky indique que ces valeurs permettent de « jouer » avec le score en fonction des connaissances sur le domaine d'application et sur l'importance des objets comparés. Dans notre cas, nous pouvons supposer que les tâches les plus importantes aient un coefficient plus grand dans cette comparaison, diminuant en conséquence la valeur du ratio si ces tâches ont beaucoup d'éléments uniques non partagés.

- Une valeur $\alpha = 0$ (resp. $\beta = 0$) permet de ne pas prendre en compte l'influence des éléments propres à T_i (resp. T_j).

4.3. ANALYSE DES TRANSITIONS : PRÉDICTION ET AIDE À LA CONCEPTION

- Définir $\alpha = \beta = 1$ donne la même importance aux deux tâches dans la comparaison.

Liens avec la taxinomie

Nous pouvons relier le score de similarité avec les cinq catégories de la dimension « contexte » de la taxinomie. Les cas de *séparation* et de *redondance totale* correspondent aux deux extrémités de l'intervalle $[0; 1]$:

Séparation : il n'y a aucun élément commun à T_i et T_j , ce qui se traduit par $f(T_i \cap T_j) = 0 \Rightarrow S = 0$.

L'implication inverse $S = 0 \Rightarrow f(T_i \cap T_j) = 0$ est également vraie, car seul le numérateur du ratio 4.1 peut être nul.

En conséquence, il y a **séparation** des tâches si et seulement si $S = 0$

Redondance totale : tous les éléments du contexte associés à T_i sont associés à T_j et inversement, ce qui se traduit par $f(T_i \cap T_j) = 1$ et $f(T_i \setminus T_j) = f(T_j \setminus T_i) = 0$ et donc $S = 1$.

Inversement, $S = 1 \Rightarrow \alpha f(T_i \setminus T_j) = -\beta f(T_j \setminus T_i)$. Or, les quatre composantes de cette équation sont positives ou nulles. En imposant $\alpha = \beta$ afin de conserver la symétrie du ratio, nous distinguons deux cas :

- soit $\alpha = \beta = 0$, ce qui donnerait un score égal à 1 quelques soient les éléments associés aux 2 tâches, cas peu intéressant ;
- soit $\alpha > 0$ et $\beta > 0$, donc $f(T_i \setminus T_j) = f(T_j \setminus T_i) = 0$ et en conséquence $S = 1$.

En conséquence, il y a **redondance totale** si et seulement si $S = 1$.

Les cas intermédiaires $0 < S < 1$ correspondent aux trois autres types de catégories (*intersection*, *réduction*, *extension*) mais la connaissance de la valeur seule de S ne permet pas de conclure sur la nature de la transition. Si on impose $\alpha = \beta$ (pour les raisons de symétrie invoquée précédemment), nous pouvons néanmoins noter les correspondances suivantes :

Réduction : les éléments associés à T_j sont tous associés à T_i (« $T_i \supset T_j$ »), on peut ainsi simplifier le calcul de S à la formule

$$S(T_i, T_j) = \frac{f(T_j)}{f(T_i \cap T_j) + \alpha f(T_i) + (1 - \alpha) f(T_j)}$$

Extension : les éléments associés à T_i sont tous associés à T_j (« $T_i \subset T_j$ »), on peut ainsi simplifier le calcul de S à la formule

$$S(T_i, T_j) = \frac{f(T_i)}{f(T_i \cap T_j) + (1 - \beta) f(T_i) + \beta f(T_j)}$$

Intersection : si l'intersection entre les deux contextes est non vide (ce qui correspondrait au cas de la séparation), on simplifie le calcul du ratio de similarité à la formule

$$S(T_i, T_j) = \frac{f(T_i \cap T_j)}{(1 - \alpha)f(T_i \cap T_j) + \alpha f(T_i \cup T_j)}$$

Notons que ces trois dernières formules ne suffisent pas à catégoriser une transition sur l'axe contextuel, il faut également y associer la comparaison des « ensembles » T_i et T_j (afin de voir si l'un est inclus dans l'autre notamment).

Intérêt pour l'analyse prédictive des scénarios

Les fondements théoriques de la *Distributed Situation Awareness* de Stanton *et al.* (2009) s'appuient sur la théorie des schémas (§ 3.3.2.3) les définissant comme la « structure active de connaissance qui guide l'exploration et l'interprétation des informations », reprenant les idées de Neisser (1976). L'anticipation des événements à venir détermine ce que nous allons faire, quelles informations nous allons chercher à acquérir, créant ainsi un cycle « perception-action » où les schémas guident l'action et sont modifiés en retour par les résultats perçus de cette action.

Ainsi, nous pouvons émettre l'hypothèse que des schémas proches vont induire des actions proches, et l'activation d'un nouveau schéma proche sera plus facile que celle d'un schéma totalement différent, pouvant induire une charge cognitive supérieure.

Nous proposons d'appliquer cette hypothèse à notre cas d'étude, où la construction du modèle de SA s'appuie sur cette théorie des schémas et la définition de schémas génotypes associés à chaque tâche de l'opérateur, schémas qui seront activés au fur et à mesure des transitions du scénario. Ainsi, pour deux tâches d'une transition considérée, l'acquisition de la SA pour la nouvelle tâche serait d'autant moins difficile, cognitivement parlant, que le contexte de la tâche (et donc le nouveau schéma activé) est proche du contexte de la tâche précédente (le schéma précédemment activé).

Le calcul du score de similarité de Tversky nous donne un aperçu de la similarité des contextes associés à chaque tâche. Dans le cadre d'une analyse prédictive de l'impact des transitions sur la *Situation Awareness* des opérateurs, nous pouvons ainsi imaginer calculer ce score pour l'ensemble des transitions du scénario considéré et définir un classement de ces transitions en fonction cette valeur.

Ce classement pourrait ensuite guider l'expert dans son analyse en définissant des priorités sur l'ordre des transitions à analyser plus en détail, celles ayant le moins d'éléments en commun nécessitant un plus grand travail de reconstruction de la *Situation Awareness* de la part de l'opérateur.

Nous notons cependant que ce travail de calcul, s'il doit être appliqué à l'ensemble des transitions d'un scénario, peut s'avérer fastidieux à réaliser « à la main ».

4.3.2 Synthèse des réflexions

Dans cette section, nous avons présenté deux pistes de réflexion portant sur la prédiction de la difficulté « cognitive » des transitions : une approche qualitative, s'appuyant sur la taxinomie définie en introduction de ce chapitre, et une approche quantitative, soutenue par le calcul d'un score de similarité entre les contextes associés aux tâches. La mise en œuvre de ces méthodes globales d'analyse permet de répondre en partie à la problématique du « *où* » (« où se situent les transitions les plus difficiles ? ») définie en introduction de ce mémoire.

Toutefois, nous privilégions dans la suite de nos travaux une approche plus « atomique » à l'analyse des transitions, abordant la problématique du « *comment* » (« comment la transition impacte l'opérateur ? »). A cette fin, nous proposons une méthodologie d'analyse « locale » des transitions, basée sur l'espace de modélisation dual défini dans le chapitre courant et sur la définition de méthodes d'analyses plus détaillées, telles que l'extraction des plus grands sous-graphes communs et l'analyse de la saillance des éléments impliqués dans une transition.

5

La méthodologie SAMANTA

Ce chapitre est consacré à la présentation de nos travaux de thèse liés au deuxième usage pour l'analyse des transitions cité dans la section 4.3 : **l'aide à conception**. A cette fin, nous proposons une méthodologie d'analyse des transitions. Ce processus d'analyse intervient dans le cadre de la conception d'un nouveau système, à partir de systèmes existants (nouvel aéronef, nouveau système de drone...) ou pour la définition d'un système ayant peu de points communs avec les systèmes existants.

Cette méthodologie met en œuvre un certain nombre de méthodes de comparaison des tâches, afin d'étudier comment une transition particulière va avoir un impact sur la *Situation Awareness* d'un opérateur, en comparant notamment les éléments de situation associés aux deux tâches impliquées dans la transition. Ces travaux reposent sur la taxinomie des transitions présentée en section 4.1 et l'espace de modélisation dual des transitions introduit dans la section 4.2.

Dans un premier temps (section 5.1), nous introduisons la méthodologie SAMANTA (*Situation Awareness Modeling & ANalysis for Transition Amelioration*) et nous en décrivons succinctement les deux phases – modélisation et analyse – chacune divisée en plusieurs étapes. Les trois parties suivantes (sections 5.2, 5.3 et 5.4) détaillent le déroulement des cinq étapes composant les deux phases de la méthodologie. Enfin, nous proposons une illustration de l'application de la méthodologie au cas d'un jeu vidéo de tir à la première personne (section 5.5).

Note sur la terminologie

La méthodologie fait intervenir deux acteurs principaux : l'*analyste* et l'*opérateur*. Nous utilisons ces deux termes dans la suite de ce chapitre en tant que termes génériques afin d'indiquer qui est en charge des actions décrites.

Le rôle d'*analyste* est tenu par la personne en charge de l'analyse et de l'intégration des résultats à la conception du nouveau système. Nous considérons que cette personne possède une certaine sensibilité (si ce n'est une sensibilité certaine voire un niveau d'expertise avancé) dans le domaine de l'ergonomie et des facteurs humains.

Le rôle d'*opérateur* décrit les utilisateurs typiques des systèmes considérés. Nous utilisons ce terme spécifique et non le mot plus générique « utilisateur » afin de spécifier qu'il s'agit d'un utilisateur expert dans l'opération du système complexe

considéré. Dans certains cas, notamment lors de la description des phases d'entretien, nous utiliserons également le terme « opérateur » afin de décrire les experts du domaine considéré (*System Matter Expert* en anglais), qui sont des sources d'informations aussi importantes et correspondent à des profils tels que des formateurs ou des ingénieurs spécialistes dans la conception de systèmes équivalents.

5.1 PRÉSENTATION DE LA MÉTHODOLOGIE

Nous introduisons dans cette section la méthodologie SAMANTA, acronyme pour « *Situation Awareness Modeling & ANalysis for Transition Amelioration* », que nous traduisons par « Modélisation et analyse de la *Situation Awareness* pour l'amélioration des transitions ». Cette méthodologie a été créée avec deux objectifs principaux :

1. Notre premier objectif est de formaliser les différentes étapes de création du modèle introduit dans la section 4.2, afin d'appuyer l'intégration de ce modèle dans le cadre d'un processus plus global d'analyse des transitions. Conformément au modèle initial DSA, ce **processus de modélisation** s'appuie sur l'extraction des données auprès des opérateurs du système et des experts du domaine considéré.

Ce processus de modélisation est décrit dans une première phase de la méthodologie, il est composé de trois étapes :

Étape 1 : extraction des connaissances

Cette première étape est inspirée de l'étape de collecte des données décrite par Stanton pour la construction du modèle DSA. Plusieurs moyens d'analyse de l'activité sont mis en œuvre : entretiens et observations des opérateurs de systèmes actuels, entretiens avec des experts du domaine, ou encore analyse de la littérature et de rapports experts. Cette étape est décrite dans la section 5.2.1 page 116.

Étape 2 : construction des modèles

Cette deuxième étape englobe la construction des deux modèles de notre approche : le modèle de tâches, décrit par la notation CTT (Paternò *et al.*, 1997) et le modèle de *Situation Awareness*, décrit par un réseau propositionnel, issu du modèle DSA (Stanton *et al.*, 2006). Nous détaillons dans les sections 5.2.2 page 117 et 5.2.3 page 128 la façon dont sont construits ces modèles, à partir des données collectées lors de l'étape 1.

Étape 3 : définition des contextes associés aux tâches

Lorsque les deux modèles ont été construits, l'analyste définit, pour chaque tâche de l'arbre CTT, les exigences en terme de *Situation Awareness*. Cette définition du contexte de SA est traduite par l'association d'un sous-ensemble d'éléments de situations – des nœuds du réseau propositionnel – à la tâche considérée. Cette étape est décrite dans la section 5.2.4 page 133.

A l'issue de cette étape, une validation des modèles construits et des associations tâches / contextes est effectuée auprès d'opérateurs et d'experts du domaine afin de les compléter, le cas échéant.

2. Notre deuxième objectif est de répondre à la problématique formulée dans l'introduction de cette thèse à savoir étudier l'impact des transitions sur la *Situation Awareness* de l'opérateur. L'étude de cet impact se traduit par l'analyse des transitions de manière locale (à l'inverse d'une analyse globale du scénario, comme nous l'avons présenté dans le chapitre précédent). Ce **processus d'analyse des transitions** s'appuie sur différentes méthodes de comparaison des contextes associés aux tâches impliquées dans une transition. Nous intégrons ce processus d'analyse à une deuxième phase de la méthodologie, décrite en deux étapes :

Étape 4 : analyse des transitions

Cette étape consiste à passer en revue les différentes transitions pertinentes du scénario considéré. A partir des scénarios types (section 4.3.1.1), l'analyste examine les transitions opérationnellement pertinentes (qui ont éventuellement pu être filtrées à travers l'une des pistes présentées dans la section 4.3.1) afin d'évaluer les impacts sur les éléments de SA de chaque transition. Afin de mesurer ces impacts, nous proposons un ensemble de méthodes de comparaison entre tâches, que nous présentons dans la section 5.3 page 138 de ce chapitre.

Étape 5 : recommandations pour la conception

A l'issue de l'analyse, l'analyste va pouvoir émettre un certain nombre de recommandations en termes d'impacts sur les interfaces. Ces recommandations s'appuient sur les résultats des méthodes d'analyse de l'étape précédente et peuvent mener, par exemple, à la définition de maquettes d'interface pour le nouveau système. Afin de soutenir ces recommandations, l'analyse pourra s'appuyer sur des guides, normes et critères existants dans le domaine des facteurs humains et de l'ergonomie.

Cette étape est intégrée en partie à la section 5.3, où l'on décrira les éléments mis en valeur par chaque méthode de comparaison. Quelques pistes de réflexion portant sur les recommandations seront également décrites dans la section 5.4 page 155.

Les étapes de la méthodologie SAMANTA sont décrites sur la figure 5.1. Dans le cadre de notre approche centrée sur les utilisateurs, nous intégrons ceux-ci à différentes étapes clés dans le processus de modélisation et d'analyse, à la fois comme une source de données en entrée de la phase de modélisation (étape 1) mais également pour la validation des associations entre éléments de situation et tâches, avant d'entamer l'analyse, ou encore à la fin du processus afin de valider les interfaces produites (à travers des tests utilisateurs par exemple).

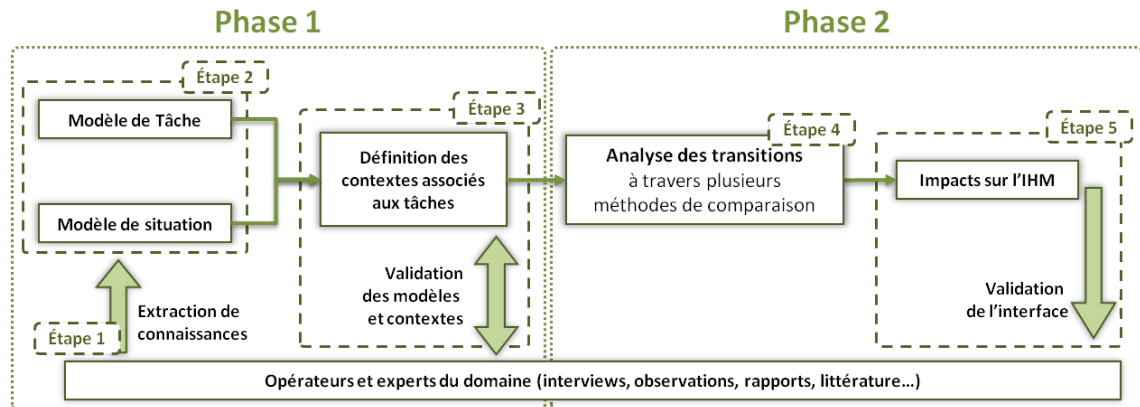


FIGURE 5.1 : description générale de la méthodologie SAMANTA

5.2 CONSTRUCTION DES MODÈLES (PHASE 1)

Nous présentons dans cette section les étapes constituant la première phase de la méthodologie SAMANTA. Ces étapes précisent la façon dont sont construits le modèle de tâches et le modèle de situation, à partir des connaissances extraites des opérateurs ou de la littérature.

5.2.1 Extraction des connaissances (étape 1)

La première étape de la méthodologie SAMANTA consiste à extraire les connaissances nécessaires à la conception du nouveau système. Cette extraction de connaissances se fait à partir de données issues de sources diverses, comme les entretiens ou observations auprès d'opérateurs et experts du domaine, ou encore la revue de la littérature et de rapports techniques décrivant des systèmes existants.

L'effort de cette thèse ne portant pas sur la définition de ces méthodes d'extraction, nous indiquons dans cette section celles qui nous paraissent les plus adéquates vis-à-vis des systèmes considérés et des modèles construits ensuite. L'annexe A apporte des compléments bibliographiques, décrivant notamment les méthodes suggérées dans cette section.

Dans le cadre de la méthodologie SAMANTA, la collecte des données doit permettre de nourrir l'étape de modélisation (étape n°2) ainsi que l'étape de définition des exigences de SA pour les tâches (étape n°3 d'association). Dans leurs travaux sur le modèle DSA, Stanton et ses collègues s'appuient sur l'analyse hiérarchique de tâches (HTA) (Annett et Duncan, 1967), à travers des observations par l'analyste ou des entretiens, suivant la méthode des décisions critiques (*Critical Decision Method*, CDM, Crandall *et al.* (2006)), décrite en annexe A.2.

Lorsque la méthodologie SAMANTA est utilisée pour la conception de système où peu d'informations sont disponibles, les connaissances peuvent être générées à

partir de scénarios prospectifs, comme le proposent Nehme *et al.* (2006) avec la méthode d'analyse cognitive des tâches hybride (*hybrid Cognitive Task Analysis*, hCTA), décrite en annexe A.3.

Ainsi, plusieurs méthodes d'analyse (entretiens, questionnaires, observations, analyse de la littérature...) peuvent être combinées afin d'obtenir des résultats complets et des points de vue différents sur le système considéré. Le choix de ces méthodes dépendra du domaine de recherche, de la facilité à accéder aux opérateurs experts (qui représentent les meilleures sources de données sur un système existant) et à leur situation de travail mais également des compétences et préférences des personnes en charge de l'analyse.

Dans ce paragraphe, nous anticipons sur les deux sous-sections suivantes où sont décrites les étapes de construction des modèles. En effet, nous souhaitons préciser les méthodes d'extraction de connaissances sur lesquelles nous nous appuyons pour faciliter cette construction. Ces choix correspondent à ceux préconisés par les auteurs des formalismes considérés.

- La construction des arbres de tâches en utilisant la notation CTT peut se reposer sur une analyse hiérarchique des tâches (HTA) afin de décrire la décomposition hiérarchique de ces tâches et sous-tâches (Annett et Duncan, 1967). Cette analyse peut être complétée par la définition de scénarios décrivant l'enchaînement temporel des tâches. Ces informations peuvent être obtenues auprès des opérateurs à travers la méthode des décisions critiques (annexe A.2), permettant notamment de construire la chronologie des événements (étape 3 de CDM) et donc un scénario, et définir les buts et sous-buts de l'opérateur lors de la prise de décision (étape 5 de CDM).
- La construction du modèle de SA sous la forme d'un réseau propositionnel s'appuie, dans les travaux de Stanton *et al.* (2009), sur la méthode des décisions critiques, en particulier l'approfondissement mené dans l'étape 5.

Dans la suite de la méthodologie, nous considérons que nous avons accès à des transcriptions d'entretiens avec les opérateurs et que les données nécessaires à la modélisation sont disponibles.

5.2.2 Construction du modèle de tâches (étape 2a)

La construction du modèle de tâches est basée sur la notation ConcurTask-Tree (pour *Concurrent Task Tree*, abrégé en CTT), introduite par Paternò *et al.* (1997). La notation CTT se présente sous la forme d'un arbre dont les nœuds représentent les tâches exécutées par l'opérateur. Chaque tâche peut être décomposée en sous-tâches, représentées comme des descendantes de ce nœud. Cette décomposition hiérarchique est complétée par une description des enchaînements temporels entre les tâches d'un même niveau hiérarchique et ayant le même ascendant direct (par

construction, chaque nœud possède un unique ascendant). Notons que l'usage de la notation CTT est appuyée par l'éditeur CTTE (Mori *et al.*, 2002) disponible en ligne ¹.

Le processus de modélisation décrit dans la présente section repose sur l'analyse hiérarchique des tâches menée au cours de l'étape 1 de la méthodologie. Paternò (2003) suggère de baser la modélisation sur des scénarios et cas d'utilisation (*use cases*) définis au cours de cette première étape. Le modèle de tâches sera construit à partir des tâches et sous-tâches mises en exergue lors de cette analyse, qu'il s'agisse de tâches existantes dans les systèmes courants ou d'expression de besoins pour les futurs systèmes.

Dans cette section, nous décrivons la façon dont nous envisageons la description du modèle de tâches pour son utilisation dans le cadre de la méthodologie SAMANTA.

5

5.2.2.1 Niveau d'abstraction des tâches considérées

La notation CTT permet de représenter les différents niveaux d'abstraction des tâches de l'activité d'un opérateur, d'un niveau d'abstraction élevé, proche de la racine de l'arbre (au sommet) jusqu'à un niveau très concret, permettant la description des interactions de l'opérateur avec le système à travers des actions « basiques » (clics, affichage de données à l'écran...).

Dans le cadre de la méthodologie SAMANTA, nous décrivons les tâches principales de façon précise sans entrer dans la description des interactions et des actions basiques. En effet, nous considérons les transitions entre tâches d'un niveau hiérarchique « élevé », pour lesquelles la définition d'exigences en termes de *Situation Awareness* est pertinente.

Ainsi, nous nous limiterons à la description des buts et sous-buts définis pour chaque phase de la mission, et ne décrirons pas la manière concrète dont ces buts peuvent être atteints. Chaque tâche est associée à un but, mais plusieurs tâches peuvent répondre au même but, offrant plusieurs moyens de l'atteindre et donc à chaque but peut être associé un ensemble de tâches. Lors de la description de la composition hiérarchique des cas d'application présentés dans ce mémoire, nous avons par exemple considéré :

- les différents modes de contrôle du système, pouvant traduire les différents niveaux d'autonomie de la machine vis-à-vis de l'opérateur (à travers la définition d'un mode manuel ou d'un mode automatique pour une tâche de pilotage de drone par exemple),

1. <http://giove.cnuce.cnr.it/ctte.html>

- les tâches courantes pour la réalisation d'une mission (surveillance constante de paramètres particuliers...),
- les tâches périphériques pouvant intervenir ponctuellement mais de manière récurrente et gêner l'exécution des tâches courantes.

Cette liste, non exhaustive, permet d'illustrer les types de tâches rencontrées dans la description du système. A partir de l'identification des buts et sous-buts de la mission, et à travers l'étude des scénarios et des *Use cases*, nous pouvons définir la décomposition hiérarchique des tâches et décrire l'enchaînement temporel prévu.

5.2.2.2 Aspect temporel de la notation

Outre la lecture hiérarchique (de haut en bas) imposée par la décomposition en tâches et sous-tâches, les arbres CTT intègrent également un axe de lecture temporel, de gauche à droite. Ces liens temporels ne concernent que les tâches de même niveau hiérarchique et partageant la même tâche parente.

La notation CTT définit 8 identificateurs² pour décrire ces liens temporels (tableau 5.1), basés sur le standard LOTOS (ISO, 1989). En termes de représentation dans l'arbre, ces identificateurs relient les tâches deux à deux et sont visualisés par un lien horizontal entre les deux tâches, situées au même niveau de décomposition hiérarchique. L'identificateur apparaît à l'intérieur d'un cercle centré sur ce lien et entre les deux tâches considérées (la figure 5.2 présente la vue de l'éditeur e-COMM (Jourde et Laurillau, 2010) pour la sélection de l'identificateur temporel entre deux tâches).

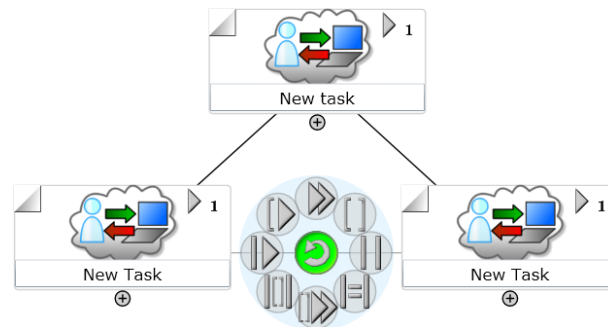


FIGURE 5.2 : sélection de l'identificateur temporel dans l'éditeur e-COMM (Jourde et Laurillau, 2010).

Cas des ambiguïtés

Nous notons que pour les identificateurs imposant un ordre dans l'exécution des

2. Nous utilisons le terme « identificateur » afin de traduire l'anglais « *operator* » et d'éviter les confusions avec notre terme « opérateur », utilisé pour nommer l'utilisateur spécialisé des systèmes complexes dynamiques considérés dans le cadre de cette thèse.

TABLEAU 5.1 : les identificateurs temporels de CTT, adapté de Paternò (2003).

Identificateur CTT	Description
[] Alternative	Un choix doit être effectué entre les deux tâches, lorsqu'une tâche est activée, l'autre devient inactive.
= Ordre indéfini	Les tâches peuvent être exécutées dans n'importe quel ordre. Mais pour commencer la deuxième tâche, il faut d'abord terminer la première.
Concurrence	Les tâches peuvent être exécutées dans n'importe quel ordre, même simultanément. Une tâche n'a pas besoin d'être terminée pour que l'autre puisse débuter.
[[]] Concurrence avec échange d'informations	Idem. Mais les deux tâches échangent des informations lors de l'exécution concurrente.
[> Désactivation	La première tâche (généralement une tâche itérative) est complètement interrompue par la seconde tâche.
> Interruption/Reprise	Cas « classique » de l'interruption : la première tâche est interrompue par la deuxième qui, lorsqu'elle se termine, permet la reprise de la première tâche dans l'état précédent.
>> Activation	La première tâche doit se terminer pour que la deuxième puisse s'exécuter, dans cet ordre (de gauche à droite).
[] >> Activation avec passage d'informations	Idem. Mais la première tâche transmet des informations à la seconde.

tâches (activation avec ou sans passage d'informations, désactivation et interruption), l'ordre des tâches est défini par l'ordre de lecture, de gauche à droite.

Plusieurs identificateurs de types différents peuvent apparaître sur une même

ligne, rendant la lecture parfois difficile, notamment lorsqu'il s'agit d'identificateurs ne prenant pas en compte le sens de lecture (alternative, ordre indéfini...). Afin de lever les ambiguïtés, Paternò *et al.* (1997) proposent deux solutions :

- la définition de priorités entre les identificateurs, basées sur les priorités définies dans le standard LOTOS (ISO, 1989). Le tableau 5.1 respecte cet ordre de priorité, de la priorité la plus haute (identificateur d'alternative []) à la priorité la plus faible (identificateur d'activation avec passage d'information [] \gg).
- la définition de tâches intermédiaires permettant de regrouper les tâches dont l'enchaînement temporel est le même sous le chapeau d'une tâche « factice », suivant en quelque sorte un principe de factorisation en fonction de l'ordre des identificateurs.

Nous avons favorisé l'usage de tâches factices dans la construction de l'arbre. En effet, bien que suffisantes pour la vérification formelle de l'enchaînement temporel des tâches, les règles de priorité n'ont pas le même impact visuel que la factorisation en tâches et sous-tâches. Même si ce processus peut ajouter quelques tâches supplémentaires, nous considérons qu'il permet de visualiser rapidement sur l'arbre et de manière plus naturelle un enchaînement temporel.

La figure 5.3 illustre ce principe : l'arbre 5.3a présente cinq tâches au même niveau hiérarchique. Les règles de priorité indiquent que l'identificateur d'alternative liant les tâches 4 et 5 est prioritaire, suivi de la concurrence entre les tâches 3 et 4 puis des séquences, décrivant l'enchaînement des trois premières tâches activées l'une après l'autre. La figure 5.3b représente la même décomposition et les mêmes enchaînements temporels en ajoutant des tâches factices afin de clarifier la lecture.

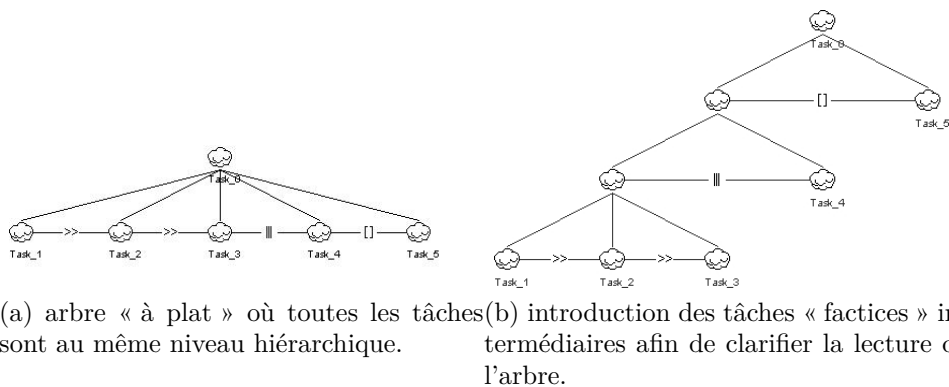


FIGURE 5.3 : illustration du principe de factorisation des tâches par l'ajout de « tâches factices » .

Bien que la version comportant les tâches « factices » ait une hauteur plus grande que la version initiale, la compréhension de l'arbre est améliorée. La nécessité pour un analyste d'introduire des tâches factices dépend de sa facilité à appréhender la notation CTT et les subtilités de l'usage des identificateurs LOTOS.

5.2.2.3 Autres informations décrivant les tâches

La notation CTT prévoit également de spécifier certaines caractéristiques individuelles, propres à chaque tâche. Celles-ci sont intégrées soit à la notation initiale (Paternò *et al.*, 1997) soit à l'éditeur CTTE et ont été ajoutées au fil des évolutions de cet outil. Dans un comparatif des outils associés aux notations de modèles de tâches, Caffiau *et al.* (2010) répertorient les différentes informations qui peuvent être ajoutées à travers l'éditeur CTTE.

Dans le cadre de la méthodologie SAMANTA, l'ajout de ces informations ne sera pas utilisé directement pour l'analyse des transitions, puisque nous nous focalisons sur l'impact en termes de *Situation Awareness* de ces changements de tâches. Néanmoins, nous avons parfois jugé pertinent d'ajouter ces informations afin d'enrichir la description des tâches et apporter un complément d'informations sur le contexte d'exécution des transitions analysées. Nous décrivons dans la suite de ce paragraphe les informations de catégories et les planificateurs locaux que nous avons utilisé.

Catégories de tâche.

Quatre catégories sont disponibles pour la classification des tâches (figure 5.4) : elles permettent de définir qui, du système ou de l'opérateur, est en charge de la tâche.

Ainsi, la catégorie *utilisateur* indique que la tâche est réalisée par l'opérateur (observation, réflexion...), la catégorie *système* indique que le système réalise la tâche (comparaison, calculs divers, affichage...), la catégorie *interaction* indique que la tâche est interactive et enfin, la catégorie *abstraite* est assignée aux nœuds de l'arbre dont les descendants n'appartiennent pas tous à la même catégorie.

Caractères itératif et optionnel.

Présentées sous la forme de cases à cocher dans l'éditeur CTTE, ces caractéristiques booléennes sont également visibles sur l'arbre CTT. Le caractère **itératif** d'une tâche est visualisé par l'ajout d'une étoile « * » près du nom de la tâche, indiquant que la tâche peut être exécutée plusieurs fois à la suite, en séquence. Son caractère **optionnel**, est quant à lui visualisé en encadrant le nom de la tâche de crochet « [] » (figure 5.4).

Ces deux identificateurs jouent le rôle d'opérations temporelles unaires, elles sont donc incluses à la table des priorités, avec la priorité la plus haute, devant toutes les opérations temporelles binaires (identificateurs LOTOS ordonnés dans le tableau 5.1).

Notons que Jourde (2011), dans sa notation COMM (variante de CTT), a regroupé ces caractéristiques sous la forme d'un identificateur unique, visualisé en exposant du nom de la tâche, et précisant la multiplicité des itérations prévues ainsi que leur type : itérations séquentielles avec l'identificateur « > », itérations parallèles avec l'identificateur « || ».

Autres planificateurs locaux.

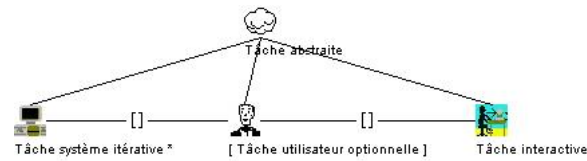


FIGURE 5.4 : représentation des catégories et caractères itératif et optionnel des tâches dans l'éditeur CTTE.

L'éditeur CTTE permet également de définir deux autres informations temporelles sur la tâche éditée : la durée et la fréquence.

- La **durée** est précisée dans un onglet spécifique de l'éditeur de tâche, et consiste à définir trois paramètres pour la performance temporelle : une durée minimum, une durée maximum et une durée moyenne. Ces paramètres se présentent comme des champs textuels où l'analyste entre un nombre entier (dont l'unité n'est pas précisée).
- La **fréquence** est définie à travers un menu déroulant, où l'analyste sélectionne l'une des trois valeurs « faible », « moyenne » ou « haute ».

D'autres informations peuvent être associées à chaque tâche à partir de l'éditeur CTTE. En particulier, l'analyste peut ajouter une description libre aux tâches, ce qui peut être intéressant pour ajouter des informations générales, ou pour noter les hypothèses posées sur certaines tâches (lors de spécifications pour des systèmes non-existants par exemple).

La possibilité est également donnée de définir des objets associés aux tâches et d'ajouter des pré-conditions sur l'état de ces objets pour la réalisation d'une tâche. Nous n'utilisons pas cette possibilité et privilégions l'utilisation d'un modèle de *Situation Awareness*, que nous jugeons plus complet.

5.2.2.4 Synthèse

La construction de l'arbre des tâches telle que nous l'avons décrite dans cette section repose sur les principes de base de la notation CTT, à savoir la décomposition hiérarchique et la description des enchaînements temporels entre tâches. Afin d'étayer cette construction initiale, suffisante à l'application de la méthodologie SAMANTA, l'analyste pourra ajouter certaines informations, portant notamment sur les aspects temporels locaux (durée, fréquence, caractères itératif et optionnel).

5.2.2.5 Discussion

Nous avons trouvé que la description des transitions à partir de la notation CTT se faisait de manière quasi-naturelle lorsque nous basions notre analyse sur des scénarios décrivant l'activité de l'opérateur, ou en s'appuyant sur des cas d'utilisation

du système considéré, comme le préconise Paternò (2003). Par exemple, pour lever le doute sur l'ordre d'apparition des tâches et choisir l'identificateur adéquat (séquence, choix, alternative, parallélisme), on peut définir, pour chaque tâche, les pré-requis à leur exécution (quelles autres tâches doivent être exécutées au préalable). De manière générale, nous pouvons traduire les descriptions associées à chaque identificateur du tableau 5.1 en questions, que l'analyste peut se poser lors de la construction de l'arbre, sur les tâches considérées, comme par exemple (liste non exhaustive) :

- « Cette tâche doit-elle être terminée pour que la deuxième puisse commencer ? »
- « Peuvent-elles être exécutées en parallèle ou doit-on faire un choix strict entre les deux ? »
- « Doit-on obligatoirement exécuter ces deux tâches à la suite l'une de l'autre (pré-requis) ? Si oui, y a-t-il un ordre prédéfini à l'exécution de ces tâches ? »
- « Cette interruption termine-t-elle la tâche ou autorise-t-elle sa reprise ? »
- « Des informations sont-elles transmises lors du passage d'une tâche à l'autre ? »

Cas des niveaux d'allocation de tâches

La notion d'allocation de tâches entre la machine et l'opérateur n'a été mentionnée que brièvement à travers la description des quatre catégories proposées par CTT. Dans notre cas d'étude, afin d'éviter l'introduction de modèles supplémentaires tels que les échelles de décisions proposées dans la méthode hCTA (annexe A.3), nous représentons ces différents niveaux d'allocation de tâches comme autant de modes de fonctionnement du système. Ainsi, pour remplir un même but, plusieurs niveaux d'allocation des tâches seront définis, décrivant autant de sous-tâches de la tâche remplissant ce but. Ces sous-tâches seront séparées par un identificateur d'alternative (chaque mode est exclusif).

Par exemple, le pilotage d'un drone peut se faire de plusieurs façons différentes. Dans le chapitre 7, le drone considéré répond à quatre modes de pilotage, qui sont autant de sous-tâches exclusives de la tâche « Piloter le drone » (section 7.4.1).

Lien avec la taxinomie des transitions

La notation CTT permet également de décrire *sur l'axe temporel* les transitions introduites dans la taxinomie. En effet, il existe une correspondance entre les huit identificateurs CTT (tableau 5.1), les 13 relations d'intervalle de Allen (décrites dans le tableau 4.1) et donc les trois catégories de la taxinomie (tableau 4.2), puisque celles-ci ont été définies à partir des relations de Allen.

Activation (A » B)

L'activation de la tâche B correspond à la fin de la tâche A, et peut donc être représentée par deux relations d'intervalles différentes : l'anachronisme et la séquence. Notons, pour l'activation, que la tâche A se produit nécessairement avant la tâche B. Il n'est donc pas possible d'utiliser les symétriques des relations d'intervalle de Allen.

Activation avec passage d'information (A []» B)

Ce cas de figure correspond, temporellement parlant, au cas précédent. Il peut donc être représenté par une relation d'anachronisme ou de séquence, sans symétrie possible.

Désactivation (A [> B)

La désactivation de la tâche A par la tâche B correspond à un seule relation d'intervalles de Allen : la séquence, où les deux intervalles s'enchaînent directement, sans « blanc » entre les deux. Cette relation n'est pas symétrique.

Alternative (A [] B)

La relation d'alternative correspond à un choix entre les deux tâches, et l'enchaînement entre ces deux tâches est possible, dans n'importe quel ordre. Nous pouvons donc la représenter par les relations d'anachronisme ou de séquence, ainsi que leurs symétriques.

Ordre indépendant (A |= B)

De la même façon, l'ordre indépendant traduit le fait que les deux tâches concernées *doivent* être réalisées, dans un ordre quelconque (alors que dans le cas de l'alternative, les deux tâches ne sont pas nécessairement exécutées pour remplir l'objectif traduit par leur tâche parente). Nous pouvons représenter l'identificateur d'ordre indépendant par l'une des relations d'anachronisme ou de séquence, ainsi que leurs symétriques.

Concurrence (A ||| B)

La relation de concurrence traduit la possibilité pour les tâches A et B de se réaliser simultanément, le début de chacune des tâches ne coïncidant pas nécessairement avec celui de l'autre. Cette relation peut donc être représentée par une relation de concomitance (ou sa symétrique), de coïncidence (ou l'une de ses variantes) ou par la relation de parallélisme, soit un total de 9 possibilités parmi les relations de Allen.

Concurrence avec passage d'information (A ||| B)

Ce cas de figure correspond, temporellement parlant, au cas précédent.

Interruption (A | > B)


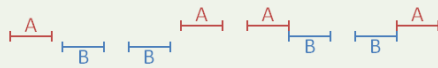

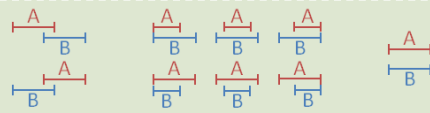
Comme son nom l'indique, la relation d'interruption ne correspond pas directement à une relation d'intervalles de Allen, mais à une combinaison de deux transitions simples. L'identificateur CTT correspondant sera donc le seul représentant de la catégorie des « interruptions » de la taxinomie.

Le tableau 5.2 synthétise les relations de Allen pouvant être utilisées afin de représenter chacun des identificateurs CTT, et attribue en conséquence à ces identificateurs une catégorie de la taxinomie (voir aussi le paragraphe 4.1.2.1).

Ainsi, les cinq premiers identificateurs CTT du tableau – l'activation (avec ou sans passage d'informations), l'alternative, l'ordre indépendant et la désactivation – sont des transitions simples. Les deux suivants – concurrence avec ou sans pas-

5.2. CONSTRUCTION DES MODÈLES

TABLEAU 5.2 : correspondance entre les identificateurs CTT et les catégories de transitions de la taxinomie.

Identificateur CTT	Opérateurs de Allen correspondants	Catégorie de transition
Activation >>	Anachronisme / Séquence – Non symétrique	Transition simple
Activation avec passage d'informations []>>		Transition simple
Alternative []	Anachronisme / Séquence – Et symétriques	Transition simple
Ordre indépendant =		Transition simple
Désactivation [>	Séquence – Non symétrique 	Transition simple
Concurrence	Concomitance / Coïncidence / Parallélisme et variantes	Ajout de tâche
Concurrence avec passage d'informations]		Ajout de tâche
Interruption/Reprise >	Combinaison de transitions simples	Interruption

sage d'informations – décrivent des ajouts de tâches. L'identificateur d'interruption correspond à l'interruption de la taxinomie.

Ces correspondances peuvent être utilisées à la fois lors de la construction de l'arbre des tâches, s'il est fait à partir d'un scénario type où sont décrites des transitions « atomiques », permettant de restreindre le choix de l'identificateur CTT associé, mais aussi en phase d'analyse des transitions (voir à ce sujet le paragraphe 5.3.2.2).

En conclusion de cette section, nous introduisons une notation formelle des arbres CTT construits, qui sera utilisée dans la suite de ce mémoire pour la description des transitions entre tâches sur l'axe temporel.

5.2.2.6 Formalisation de l'arbre de tâches

Dans la théorie des graphes, la décomposition hiérarchique définie par la notation CTT correspond à la notion d'**arborescence** (Fournier, 2006). Une arborescence est définie comme un graphe acyclique orienté possédant une racine unique et construite telle que tous les nœuds, sauf la racine, aient un unique parent et zéro ou plusieurs nœuds descendants. Deux catégories de sommets sont distingués dans ce type d'arbre :

- Les **feuilles** sont les sommets ne possédant pas de descendants, elles sont situées en bas de l'arbre ;
- Les **nœuds** correspondent à tous les autres sommets, ayant un parent unique et au moins un descendant.

Notons que dans notre cas de figure, le nombre de nœuds descendants n'est pas limité, à l'inverse des arbres *n-aires*, limitant à n descendants la décomposition de chaque nœud (les plus courants étant les arbres binaires).

Lorsque cela sera nécessaire dans la suite de ce chapitre, nous utiliserons les notations spécifiées dans cette section :

- On notera A l'arbre considéré et T_0 sa racine.
- Chaque nœud de l'arbre sera indicé en fonction de l'indice de ses ascendants et de son ordre parmi les nœuds partageant le même parent direct, en respectant l'ordre de lecture de gauche à droite.

Ainsi, pour le premier niveau, les nœuds seront notés T_1, T_2, \dots, T_n . Pour chaque niveau suivant, les nœuds conservent l'indice de leur parent auquel est concaténé leur numéro d'ordre dans le groupe des descendants (ce numéro d'ordre est réinitialisé à chaque fois que l'on change de nœud parent). La figure 5.5 présente un exemple de notation des nœuds d'un arbre à deux niveaux.

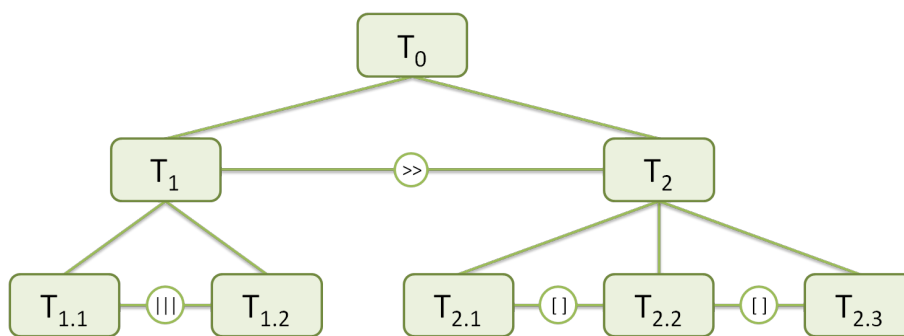


FIGURE 5.5 : principe de notation des nœuds d'un arbre enraciné.

Enfin, la particularité de la notation CTT repose dans la représentation de l'enchaînement temporel entre tâches d'un même niveau hiérarchique sous la forme de liens marqués par un identificateur temporel (tableau 5.1). Ce lien est défini par un couple de tâches liées par un identificateur temporel.

On notera par un triplet $(T_a, \textit{identificateur}, T_{a+1})$ les transitions visuellement représentées sur l'arbre A , où T_a et T_{a+1} sont deux tâches successives de même niveau hiérarchique et partageant le même parent, et *identificateur* correspond à l'un des identificateurs LOTOS définis dans la première colonne du tableau 5.1.

On notera T l'ensemble des transitions représentées dans l'arbre A : un arbre CTT sera ainsi défini par le couple (A, T) .

Par exemple, l'arbre présenté dans la figure 5.5 sera formalisé par le couple (A, T) défini tel que :

$$A = \{T_0, T_1, T_2, T_{1.1}, T_{1.2}, T_{2.1}, T_{2.2}, T_{2.3}\} \text{ et}$$

$$T = \{(T_1, \gg, T_2), (T_{1.1}, |||, T_{1.2}), (T_{2.1}, [], T_{2.2}), (T_{2.2}, [], T_{2.3})\}$$

Nous n'intégrons pas à ce formalisme les différentes informations supplémentaires, décrite dans le paragraphe 5.2.2.3, non utilisées dans le cadre de la méthodologie SAMANTA. Nous remarquons enfin, que dans le cas de l'application de cette notation à l'étude des transitions plus générale, la notation en triplet $(T_a, \textit{identificateur}, T_{a+1})$ pourra être étendue à une notation plus générale :

$$(T_a, \textit{identificateur}, T_b)$$

afin de prendre en compte les transitions possibles dans l'arbre mais non visuellement représentées (nous détaillons ces problématiques dans la section 5.3.1).

5.2.3 Construction du modèle de SA (étape 2b)

Le modèle retenu pour la description de la *Situation Awareness* des opérateurs est issu du modèle *Distributed Situation Awareness* de Stanton *et al.* (2006). Il est présenté sous la forme d'un graphe orienté, ou « réseau », construit à partir des données collectées lors de l'étape d'extraction de connaissance.

La méthode de construction de ces réseaux propositionnels ayant été présentée dans notre état de l'art (paragraphe 3.3.2.2), nous en rappelons seulement les principaux points dans cette section, en nous focalisant notamment sur l'analyse des données issues de l'étape 1 et sur la fusion de ces données afin de créer un réseau propositionnel « global », permettant de modéliser l'ensemble des éléments de SA du système considéré.

5.2.3.1 Construction des réseaux propositionnels

Dans notre modèle, la SA est représentée par un ensemble d'éléments d'information (que l'on nommera aussi « éléments de situation », traduction mot à mot de l'anglais « *situation elements* ») liés entre eux. Ce processus de cartographie de concepts (*concept mapping*) est également utilisé comme un outil pour l'analyse

cognitive de tâches car « il permet de représenter la connaissance d'une façon compréhensible par d'autres personnes, et utilisable par elles » (Crandall *et al.*, 2006, chapitre 4).

Plus que de simples nœuds ajoutés de manière aléatoire à un réseau déjà existant, le principe de construction mis en avant par Stanton *et al.* (2006) est celui du réseau propositionnel, déjà utilisé par Anderson (1989) pour représenter l'activation d'éléments de la mémoire et leur effet sur les éléments adjacents.

Le processus de construction, présenté en détail dans le paragraphe 3.3.2.2, consiste à analyser les différentes sources d'informations disponibles (transcriptions des observations et entretiens, manuels d'opération, littérature sur le domaine, rapports d'experts) et d'en extraire les éléments d'information ainsi que les liens entre ces éléments. Cette extraction consiste en « une simple analyse de contenu » (Salmon *et al.*, 2010) des documents à la recherche de mots clés et des liens causaux entre les concepts correspondant (*pilote « engage » cible, cible « suit » cap...*).

En sortie de ce processus, un réseau dit « propositionnel » est construit : les nœuds représentent les différents concepts et les liens entre ces nœuds correspondent aux liens causaux détectés, étiquetés par le verbe de relation entre les nœuds. La figure 5.6 illustre le concept d'analyse de transcription d'entretien pour en extraire un réseau propositionnel, dans le cadre d'une étude sur le combat terrestre (Salmon *et al.*, 2010).

Le processus d'extraction et de construction de réseaux propositionnels génère plusieurs réseaux, donnant ainsi accès à différents schémas phénotypes (§ 3.3.2.3), particularisation d'un schéma plus global, appelé le schéma « génotype ». Les réseaux particuliers décrits ne correspondent pas forcément à des tâches uniques mais à des ensembles de tâches, et peuvent varier selon les opérateurs interrogés ou les sources de données utilisées.

Ainsi, afin d'homogénéiser les différentes données collectées, et permettre dans un second temps la définition des éléments de situation nécessaires à l'exécution de chaque tâche, nous proposons de construire un réseau global à partir des divers réseaux construits pour chaque tâche décrite par l'opérateur.

5.2.3.2 Construction d'un réseau global

La construction d'un réseau propositionnel global finalise la construction du modèle de SA utilisé dans la méthodologie SAMANTA. En effet, en fonction du nombre de sources utilisées, de leurs types, mais également si plusieurs analystes ont participé à l'extraction et l'analyse des données, les réseaux obtenus peuvent présenter des divergences. Par exemple, un même élément d'information peut être nommé différemment dans plusieurs réseaux, ou le lien entre deux éléments peut être étiqueté différemment car deux opérateurs n'ont pas utilisé la même relation pour décrire ce lien.

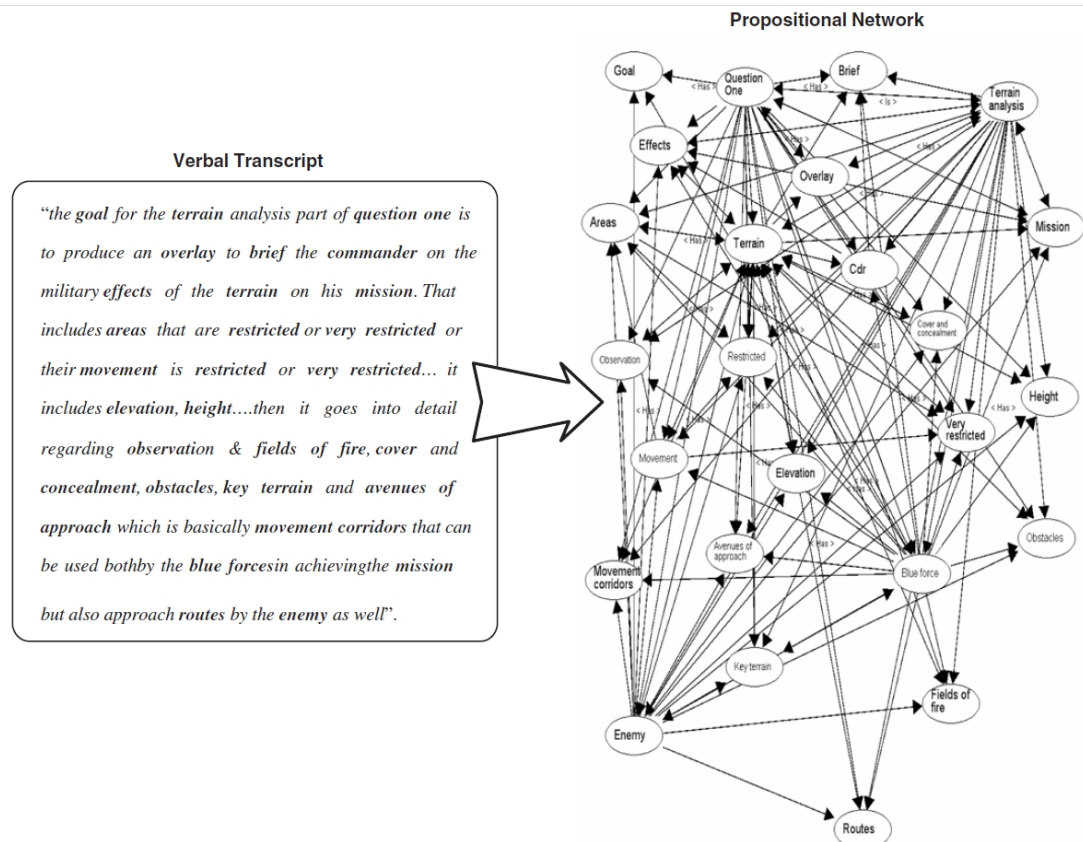


FIGURE 5.6 : exemple de création d'un réseau propositionnel à partir d'une transcription d'entretien (extrait de Salmon *et al.*, 2010).

Ainsi, nous envisageons deux stratégies pour créer un réseau global contenant l'ensemble des éléments de situation pertinents pour la réalisation de la mission, en fonction des ressources attribuées à l'étude et des choix effectués en amont :

- Si un seul analyste travaille à la fois à l'analyse des données, les nouveaux éléments peuvent être ajoutés au fur et à mesure à un seul réseau, en adaptant les libellés des nœuds et les étiquettes des liens en fonction de celle jugée la plus pertinente.
- Si plusieurs réseaux propositionnels ont déjà été construits, ceux-ci doivent être fusionnés et homogénéisés. Cette tâche, réalisée par un seul analyste, suit le même principe de sélection d'un terme unique pour chaque concept.

Dans les deux cas, nous pensons qu'il est nécessaire de fixer des termes uniques pertinents assez tôt dans le processus de modélisation et de s'y maintenir afin d'assurer l'unicité des nœuds représentant une même information.

Concernant les relations entre les nœuds et l'étiquetage de ces relations, il peut arriver que deux relations opposées (entre deux nœuds, l'une représentée par une flèche dans une direction et l'autre dans l'autre direction) soient présentes. Dans ce

cas, nous préférons simplifier la visualisation du réseau si cela est possible :

- en supprimant l'une des deux relations (par exemple, si les relations sont « représente » et « est représenté par », en supprimant la forme passive) ;
- ou en fusionnant ces deux relation en une flèche à deux extrémités dont l'étiquette permet de décrire la relation dans les deux sens (par exemple, *opérateur interagit avec* *artéfact*).

Que ce soit pour la construction des réseaux propositionnels « individuels » ou la construction du réseau global, le processus peut être facilité par l'usage d'outils informatiques supportant l'édition de graphes. Nous utilisons l'éditeur *yEd Graph Editor* gratuit, non libre, et disponible en ligne³ qui permet d'éditer des graphes orientés et possède des algorithmes permettant de ré-agencer les graphes construits afin d'en optimiser la lecture.

Notons que le réseau global construit n'est pas forcément « connexe » : certains éléments d'information ou groupes d'éléments peuvent ne pas être liés aux autres réseaux, créant ainsi des « grappes » d'éléments d'information, mais dont l'unicité de représentation a été vérifiée par l'analyste. Lorsque nous mentionnons le réseau propositionnel global (ou modèle de SA), nous considérons l'ensemble des ces réseaux partiels, s'ils existent.

L'usage de graphes pour la description des contextes associés aux tâches composant une transition nous permet de proposer des méthodes originales de comparaison des ces tâches et d'analyse des transitions considérées. Afin de soutenir ces méthodes faisant appel à des algorithmes de graphe (extraction de sous-graphes, analyse de la saillance des éléments, voir la section 5.3), nous proposons dans le paragraphe suivant de formaliser la notation du réseau propositionnel global en nous basant sur la théorie des graphes.

5.2.3.3 Formalisation du réseau propositionnel

Dans la théorie des graphes, le réseau propositionnel global correspond à la notion de **graphe orienté strict**, généralement sans **arcs opposés** (Fournier, 2006, chapitre 4).

Le graphe est défini par deux ensembles finis :

- un ensemble fini de **sommets** (*vertices* en anglais), ou **nœuds** (*nodes* en anglais) ;
- un ensemble fini d'**arcs** ou **arêtes orientées** (*edges* en anglais), reliant les sommets deux à deux.

Les réseaux propositionnels sont des arbres orientés **stricts**. Cette notion indique qu'il n'existe pas d'arcs multiples (un seul arc relie deux sommets dans le même sens)

3. <http://www.yworks.com/>

ni de cycles (les arcs partant d'un sommet pour y revenir sont prohibés). La définition d'**arcs opposés** correspond à la définition de deux arcs liant deux sommets dans un sens et dans l'autre. Comme nous l'avons précisé dans le paragraphe précédent, nous souhaitons limiter cette possibilité afin de simplifier la lecture du réseau global.

Lorsque cela sera nécessaire dans la suite de ce chapitre, nous utiliserons les notations spécifiées dans cette section. Ainsi, on notera :

- $V = \{v_1, \dots, v_n\}$ l'ensemble des sommets du graphe considéré ;
- E l'ensemble des arcs du graphe, représentés sous la forme de couples $(v_i, v_j) \in V^2$ avec $0 < i, j \leq n$;
- $L = \{l_{i,j}, 0 < i, j \leq n, (v_i, v_j) \in E\}$ l'ensemble des libellés des arcs (v_i, v_j) . Lorsque deux arcs opposés existent, nous définissons deux étiquettes symétriques (comme pour les nœuds v_5 et v_7 sur la figure 5.7)
- $G = (V, E)$ le graphe considéré.

La figure 5.7 présente un exemple d'application de la notation pour un graphe G comportant 9 nœuds et 10 arcs. Ainsi, l'arbre G de cette figure sera formalisé par le couple (V, E) tel que :

$$V = \{v_i, 1 \leq i, j \leq 9\} \text{ et}$$

$$E = \{(v_1, v_3), (v_1, v_8), (v_1, v_9), (v_2, v_1), (v_5, v_1), (v_5, v_3), (v_5, v_4), (v_5, v_7), (v_7, v_5), (v_7, v_6)\}$$

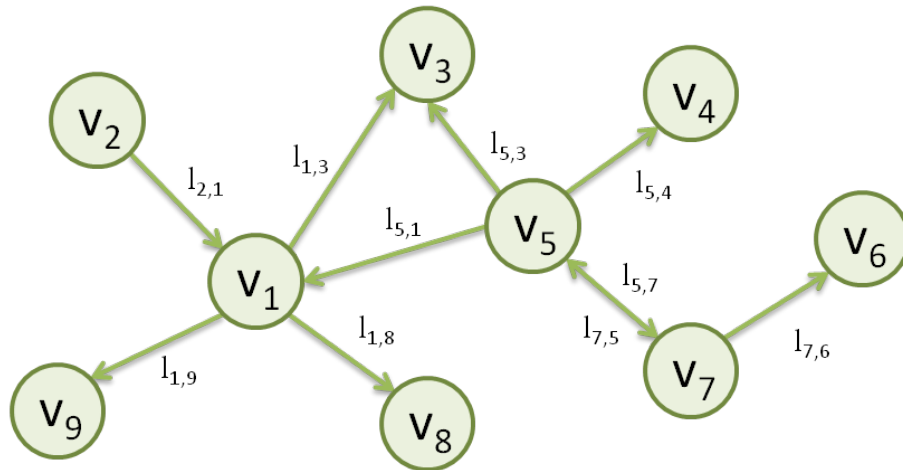


FIGURE 5.7 : illustration de la notation associée aux réseaux propositionnels.

Cette notation formelle sera associée à la formalisation de la description des arbres de tâches (§ 5.2.2.6) dans la suite de ce chapitre, afin notamment de spécifier le principe d'association des éléments de situation aux tâches puis de décrire les algorithmes mis en œuvre dans la phase d'analyse des transitions (section 5.3).

5.2.4 Définition des contextes associés aux tâches (étape 3)

La troisième étape de la méthodologie SAMANTA a pour objectif de définir précisément les éléments d'informations relatifs à la situation et nécessaires à l'opérateur pour la réalisation de la tâche considérée. Ces éléments, liés à la construction de la *Situation Awareness* de l'opérateur pour cette tâche, définissent le **contexte** associé à la tâche.

Dans la section courante, nous présentons en détail ce principe d'association entre tâches de l'arbre CTT et contextes, définis comme des sous-ensembles du réseau propositionnel global.

5.2.4.1 Association des éléments de situation aux tâches

La méthode de construction des réseaux propositionnels à partir des entretiens et des scénarios, décrite dans la section précédente, permet à l'analyste d'obtenir un premier aperçu des éléments de situation exigés pour la réalisation de chaque tâche, puisque le réseau global s'appuie notamment sur les réseaux définis dans la description des différentes tâches d'un opérateur.

Ainsi, afin de définir les contextes associés à chacune des tâches de l'arbre, nous proposons plusieurs méthodes, dont le choix dépend des données dont dispose l'analyste.

- Un point de départ intéressant est l'utilisation des scénarios et données collectées lors des entretiens avec les opérateurs. En effet, dans la section précédente, nous avons présenté comment ces données, en définissant des réseaux partiels associés au déroulement d'un scénario, permettent de construire un graphe global. Nous pouvons, pour la définition des contextes, effectuer l'opération inverse, et utiliser ces réseaux « locaux » comme points de départ pour la définition du contexte d'une tâche.
- Afin de compléter chaque contexte, l'analyste peut par exemple se poser la question « Quels sont les éléments exigés par l'opérateur pour réaliser cette tâche spécifique ? ». Les éléments du graphe répondant à cette question constituent une base solide pour la définition du contexte.

Dans leur méthode *Goal-Direct Task Analysis* (GDTA), Endsley *et al.* (2003) reprennent ce principe en extrayant, pour chaque but ou sous-but de l'analyse hiérarchique des tâches, les éléments nécessaires à leur accomplissement.

Ainsi, en s'appuyant sur ces données et son expertise, l'analyste peut obtenir une première version des contextes associés aux tâches, qu'il va définir directement sur le graphe de situation global.

5.2.4.2 Visualisation et formalisation des contextes

Sur le réseau propositionnel global, le contexte associé à une tâche constitue un sous-ensemble des nœuds et arcs du graphe. Comme pour le modèle DSA (Stanton

et al., 2006), nous proposons de visualiser cette association entre tâche et contexte en coloriant les éléments du graphe associé à la tâche considérée. Ainsi, à chaque tâche est associé un sous-ensemble colorié du graphe global.

Respectant les notations introduites dans les sections précédentes, nous notons $G_a = (V_a, E_a)$ le sous-graphe de $G = (V, E)$ représentant le contexte associé à la tâche T_a . Par définition du sous-graphe, on a $V_a \subset V$ et $E_a \subset E$.

Nous remarquons que la définition d'un contexte est faite à travers l'association d'éléments de situation, c'est-à-dire uniquement la définition d'un sous-ensemble V_a de V . La définition du sous-ensemble E_a est déduite en prenant l'ensemble des arcs de E qui ont leurs extrémités dans V_a . En effet, lors de la sélection de ce sous-ensemble, il ne serait pas justifié de supprimer un arc défini dans le réseau propositionnel global s'il relie deux éléments de situation du contexte associé à T_a . Cette remarque permet de simplifier la définition d'un contexte en construisant uniquement l'ensemble E_a par rapport à la définition générale (Fournier, 2006) selon laquelle les arcs du graphe principal E qui relient des éléments de V_a ne sont pas systématiquement ajoutés à E_a .

La figure 5.8 représente visuellement un exemple d'association entre une tâche $T_{1.2}$ et un sous-ensemble $V_{1.2} = \{v_1, v_2, v_5\}$ des sommets du graphe présenté, mise en évidence par la coloration saumon des nœuds et arcs du contexte.

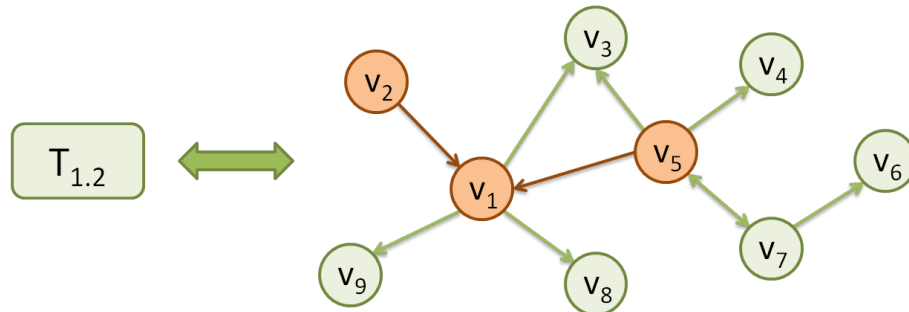


FIGURE 5.8 : association des éléments de situation à une tâche feuille.

5.2.4.3 Introduction du principe d'agrégation des contextes

Dans les systèmes complexes considérés dans le cadre de cette étude, le nombre de tâches décrites dans l'arbre CTT peut être très élevé, rendant fastidieuse et parfois complexe la définition des contextes pour chacune de ces tâches. De plus, une partie des tâches est « factice » et n'existe que pour permettre de lever les ambiguïtés sur l'enchaînement temporel des tâches. Mais ces tâches de niveaux d'abstraction élevés peuvent avoir un sens du point de vue de l'utilisateur, car elles regroupent un ensemble de sous-tâches ayant des buts connexes.

Par exemple, pour un pilote de chasse, nous pouvons imaginer une tâche de « gestion de l'environnement » qui regrouperait trois sous-tâches concurrentes : l'« orientation spatiale vis-à-vis des éléments statiques », la « gestion de la météo » et la « prise en compte des entités mobiles amies alentour ». Chaque sous-tâche décrira alors des buts légèrement différents mais permettant à l'opérateur de définir sa position et de se projeter par rapport à cette position, en fonction des éléments associés à chaque sous-tâche citée. Dans ce cas, l'analyste n'a peut-être pas défini de contexte pour la tâche « mère », mais a précisé les contextes des tâches filles.

Malgré cette absence de contexte défini pour cette tâche, elle peut être impliquée dans l'analyse d'une transition. Nous proposons deux solutions pour intégrer cette tâche sans contexte à l'analyse.

- Soit l'analyste revient aux « fondamentaux » et cherche, dans sa base de données, les éléments de situation nécessaires à la réalisation de cette tâche.
- Soit il utilise les contextes associés aux tâches « filles » de la tâche considérée, s'appuyant sur les informations associées à ces tâches pour définir le contexte de leur tâche mère.

Lorsque la première solution n'est pas pertinente, c'est-à-dire que la tâche considérée est utilisée pour « factoriser » ses sous-tâches et, nous proposons de pallier cette absence d'association en fixant des règles d'agrégation des contextes associées aux tâches « filles », afin de définir le contexte de la tâche « mère ».

Pour définir ce contexte de remplacement, nous combinons, par des opérations ensemblistes (union et intersection), les contextes associés aux sous-tâches de la tâche considérée. Ces opérations sont définies par les identificateurs temporels liant les sous-tâches entre elles.

Cas de l'union

Lorsque les sous-tâches sont concurrentes (cas de notre exemple du pilote de chasse) ou lorsque l'identificateur CTT indique une transmission d'informations entre les sous-tâches, cela signifie que l'opérateur doit connaître les éléments de situation associés à toutes ces tâches en même temps. Dans ce cas, le contexte associé à la tâche parente est défini par l'union des sous-graphes associés à chacune des tâches en jeu.

Sur la figure 5.9, afin de définir G_1 – le contexte associé à la tâche T_1 – nous considérons l'union des graphes $G_{1.1}$ et $G_{1.2}$, associés aux tâches concurrentes $T_{1.1}$ et $T_{1.2}$.

Cas de l'intersection

Dans les autres cas, lorsque les sous-tâches de la tâche sans contexte sont séquentielles, ou exécutées alternativement (alternative ou ordre indéfini), ou encore dans le cas d'une interruption, la seule certitude en termes d'éléments de *Situation Awareness* pertinents concerne les éléments communs à ces tâches : si toutes les tâches « filles » partagent un certain nombre d'éléments de situation, ceux-ci sont alors associés à la tâche « mère ». Dans ces cas de figure, nous

définissons le contexte associé à la tâche parente comme l'intersection des sous-graphes associés à ses sous-tâches.

Sur la figure 5.9, le graphe G_2 est ainsi défini comme l'intersection des trois graphes $G_{2.1}$, $G_{2.2}$ et $G_{2.3}$: un seul élément de situation est commun à ces trois tâches et est ainsi assuré d'être associé à la tâche T_2 .

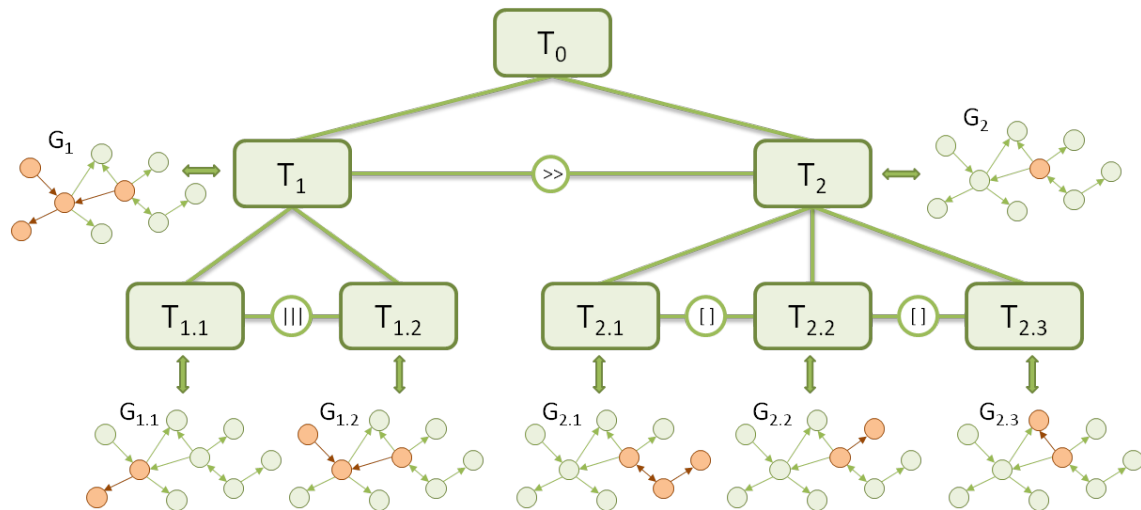


FIGURE 5.9 : illustration de l'association tâche / contexte pour l'ensemble des tâches d'un arbre, en appliquant le principe d'agrégation aux tâches T_1 et T_2 . Aucun contexte n'est défini pour la tâche racine.

Le tableau 5.3 synthétise les opérations ensemblistes utilisées pour créer le contexte associé à une tâche sans contexte, en fonction des identificateurs CTT séparant les sous-tâches « filles » de la tâche sans contexte.

Dans le cas où différents identificateurs seraient utilisés, nous appliquons les opérations ensemblistes sur les sous-graphes associés dans l'ordre des priorités associées aux identificateurs. Celles-ci correspondent à l'ordre d'apparition des identificateurs dans le tableau 5.3 (de la priorité la plus haute, pour l'alternative, à la priorité la plus basse pour l'activation avec passage d'information).

Tous les sous-graphes définissant les contextes associés aux sous-tâches étant définis à partir d'un même graphe G , nous notons que leurs unions et intersections sont également des sous-graphes de G . Enfin, notons que cette agrégation peut être appliquée de manière récursive : si aucun contexte n'est associé à une sous-tâche de la tâche sans contexte, on pourra appliquer les mêmes règles d'agrégation sur cette sous-tâches, et ainsi de suite, jusqu'aux tâches feuilles de l'arbre.

Discussion sur ce principe d'agrégation

La mise en œuvre du principe d'agrégation peut se faire de manière automatisée dans un système qui appuierait l'utilisation de la méthodologie SAMANTA (c'est

TABLEAU 5.3 : correspondance entre les identificateurs CTT et les opérations ensemblistes pour la combinaison de contextes.

Identificateur CTT	Opération
[] Alternative	Intersection
= Ordre indéfini	Intersection
Concurrence	Union
[[]] Concurrence avec échange d'informations	Union
[> Désactivation	Intersection
> Interruption/Reprise	Intersection
>> Activation	Intersection
[] >> Activation avec passage d'informations	Union

le cas de l'outil développé dans le cadre de cette thèse et présenté dans le chapitre 6). De plus, nous pensons que cette méthode peut permettre de pallier les absences de contextes définis pour les tâches les plus abstraites, notamment lorsqu'elles agrègent un ensemble de tâches concurrentes, dont les contextes vont être fusionnés, ou lorsque l'analyste ne sait pas quels éléments associer à ces tâches. Dans ce cas, ce principe permet de fonctionner en « mode dégradé » pour l'analyse de ces tâches (et l'analyste doit en être conscient).

Nous avons mentionné le fait que ce principe d'agrégation est récursif. Cependant, cet algorithme récursif impose la définition d'un contexte pour tous les descendants de la tâche considérée à un niveau donné. En effet, si un des contextes rencontrés lors du parcours des descendants de la tâche « mère » n'est pas défini, l'opération d'union ou d'intersection avec ce contexte perd tout son sens. Nous pensons que rien ne se substitue à l'expertise des analystes et les informations obtenues par le biais de l'agrégation doivent ainsi faire l'objet d'une attention particulière.

5.2.5 Synthèse de la première phase

Cette section nous a permis de décrire les trois premières étapes de la méthodologie permettant de construire un modèle de tâches et un modèle de *Situation Awareness*, puis de définir les liens entre ces deux modèles. Ce processus de modéli-

sation et de définition des éléments de situation associés à une tâche, définis comme des « exigences informationnelles » pour la construction d'une *Situation Awareness* optimale par l'opérateur, est crucial pour la suite de l'analyse. Notons qu'il peut également être appliqué dans le cadre de processus de conception plus généraux adoptant une approche centrée sur les utilisateurs.

Afin de finaliser cette phase de modélisation et de valider, voire compléter, les modèles et contextes définis, il nous semble important de les soumettre à l'expertise d'opérateurs ou experts du domaine, par l'intermédiaire d'entretiens, en groupe ou individuel. Les modifications proposées par les opérateurs lors de cette validation doivent, le cas échéant, être intégrées aux modèles avant de passer à la deuxième phase de la méthodologie SAMANTA, l'analyse des transitions.

5.3 ANALYSE DES TRANSITIONS (PHASE 2, ÉTAPE 1)

La phase d'analyse des transitions constitue le cœur de la méthodologie SAMANTA et son objet principal. Cette phase se décompose en deux étapes liées :

- une étape d'analyse à proprement parler, où sont passées en revue les transitions pertinentes des scénarios types considérés afin d'évaluer les impacts éventuels sur les éléments de *Situation Awareness* de chaque transition ;
- une étape consistant à émettre des recommandations sur les futures interfaces du système étudié, prenant en compte les résultats de l'analyse et intégrant notamment les éléments de *Situation Awareness* afin d'appuyer, dans la transition entre tâches, la construction par l'opérateur, d'une SA adéquate.

Dans l'ensemble de cette section, nous considérons que l'analyste étudie une transition spécifique, entre deux tâches, extraite du scénario type. Cette transition « atomique » est définie par deux tâches distinctes, auxquelles sont associés deux contextes (comme décrite dans la section 5.2.4, décrivant l'étape 3 de la méthodologie SAMANTA).

Ces deux tâches (et les contextes associés), sont liées par un unique identificateur CTT. Lorsque ce lien n'est pas visible directement dans l'arbre CTT, il sera défini à partir des liens entre les tâches voisines et/ou les tâches ascendantes, selon l'algorithme proposé dans la sous-section suivante.

Lorsqu'elle ne sera pas illustrée par un exemple concret, on notera $(T_a, \text{identificateur}, T_b)$ la transition considérée, et G_a & G_b les sous-graphes décrivant les contextes associés.

Les sous-sections subséquentes décrivent différentes méthodes de comparaison des tâches et contextes associés, pour l'analyse de l'impact des transitions sur la SA de l'opérateur. Nous intégrons à la présentation de ces méthodes des éléments de réflexion sur l'amélioration possible des interfaces à partir des analyses produites. Nous nous appuierons en particulier sur les principes de conception proposés par

Endsley *et al.* (2003) et intégrant les réflexions sur l'intégration de la SA lors de la conception des systèmes.

5.3.1 Préambule : les transitions dans le modèle de tâches

Dans le chapitre précédent (§ 4.3.1.1), nous avons introduit le concept de scénarios types, définis lors de l'analyse de l'activité des opérateurs, qui décrit la suite de transitions « atomiques » et les tâches impliquées dans les transitions. L'analyse de ces transitions est effectuée à partir de leur identification sur l'arbre CTT et la comparaison des tâches associées.

La visualisation graphique de l'arbre CTT impose certaines limites en termes de représentation des transitions, notamment dues à la décomposition hiérarchique. En effet, le nombre de transitions présentées ne correspond pas à toutes les transitions possibles dans l'arbre : certaines tâches non voisines sur l'arbre vont parfois se suivre, s'exécuter en parallèle ou s'interrompre dans la réalité.

Selon la description du scénario et la construction de l'arbre, il peut être nécessaire d'analyser une transition entre ces deux tâches qui ne sont pas d'immédiates voisines, et il est alors nécessaire de définir l'identificateur CTT reliant les deux tâches. Le problème revient alors à déterminer ce que représente la variable *identificateur* dans le triple $(T_a, \text{identificateur}, T_b)$. Nous relevons deux cas de figure, présentés dans les paragraphes suivants.

5.3.1.1 Tâches de même niveau hiérarchique, au voisinage non immédiat

Une transition peut impliquer deux tâches de même niveau hiérarchique mais qui ne sont pas d'immédiates voisines (figure 5.10). Ces tâches ayant le même parent direct peuvent donc être reliées entre elle par une suite de transitions et d'identificateurs CTT. Nous proposons l'algorithme de recherche de l'identificateur suivant :

1. Dans un premier temps, nous déterminons la suite de transitions reliant les deux tâches T_a (tâche la plus à gauche) à T_b (tâche la plus à droite). Cette suite est définie par l'ensemble des triplets $(T_i, \text{identificateur}, T_j)$ rencontrés en lisant le niveau hiérarchique de gauche à droite.
Par exemple, dans le cas de la figure 5.10, la tâche la plus à gauche est $T_{2.1}$, la tâche de droite est $T_{2.3}$ et la suite permettant de lier les deux tâches est $\{(T_{2.1}, [], T_{2.2}), (T_{2.2}, [], T_{2.3})\}$, sous-ensemble de l'ensemble des transitions T .
2. Il s'agit ensuite d'extraire, de ce sous-ensemble de transitions, l'identificateur CTT ayant la priorité la plus haute (telle que définie dans la section 5.2.2, tableau 5.1). Cette priorité déterminera alors l'identificateur pour la transition considérée.
Ainsi, dans notre exemple, les deux identificateurs étant les mêmes (ce qui est généralement le cas lorsque l'arbre CTT a été construit de façon à limiter

les ambiguïtés), on définit la nouvelle transition $(T_{2.1}, [], T_{2.3})$. Notons que la représentation de l'identificateur CTT associé sur l'arbre a été arbitrairement définie à celui le plus à gauche (identificateur entouré sur la figure).

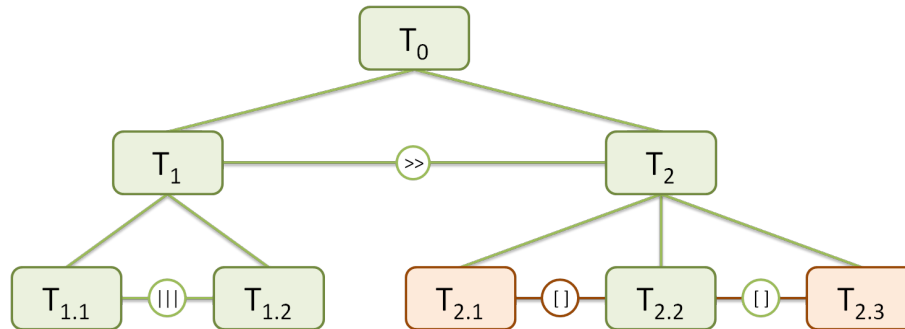


FIGURE 5.10 : transition entre deux tâches non voisines ayant le même ascendant direct.

5

5.3.1.2 Tâches n'ayant pas de parent direct commun

Le deuxième type de transition correspond aux transitions engageant deux tâches de parents directs différents (figure 5.11). Dans ce cas de figure, l'algorithme de recherche de l'identificateur CTT définissant la transition est décomposé en deux phases :

1. Pour chacune des deux tâches considérées, on recherche dans l'ensemble des leurs ascendants respectifs (les tâches permettant de « remonter » à la racine de l'arbre), les deux tâches ascendantes ayant leur tâche parente commune. L'unicité de ce parent commun est assurée par la définition de l'arbre (et l'assurance de la définition, pour chaque nœud de l'arbre, d'un parent unique). Dans notre exemple, figure 5.11, nous cherchons le parent commun aux tâches $T_{1.2}$ et $T_{2.2}$. Il s'agit de T_0 , les deux tâches ascendantes considérées sont donc T_1 et T_2 .
2. Nous nous retrouvons donc dans le premier cas de figure, de deux tâches voisines, mais dont le voisinage n'est pas forcément immédiat.
 - Si ces deux tâches sont reliées par un identificateur CTT (dans notre exemple, par exemple, T_1 et T_2 sont directement liées), celui-ci est défini comme l'identificateur pour la transition analysée ;
 - Sinon, on applique l'algorithme de recherche d'identificateur présenté dans le premier cas de figure (paragraphe 5.3.1.1).

5.3.2 Statistiques générales sur la transition

L'analyse d'une transition débute par la comparaison des deux tâches impliquées, et particulièrement des contextes associés à ces deux tâches. Dans un premier temps,

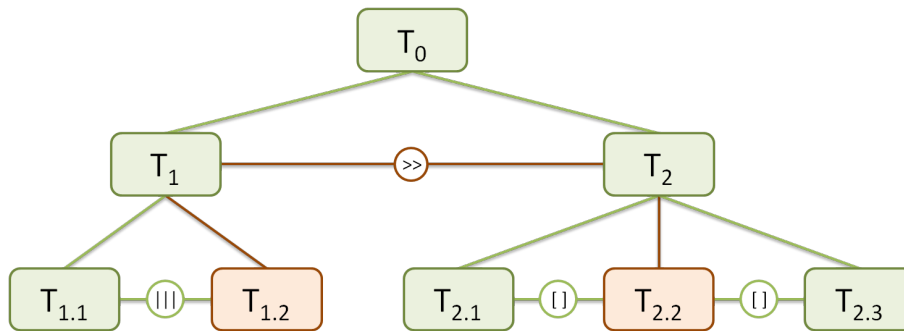


FIGURE 5.11 : transition entre deux tâches non voisines ne partageant pas le même parent.

nous décrivons les comparaisons effectuées sur les éléments de situations associés à chacune de ces tâches, sans tenir compte de la modélisation en réseau propositionnel. Nous abordons également l'aspect temporel des transitions, en faisant un lien avec la taxinomie définie dans le chapitre précédent.

5.3.2.1 Similarité entre tâches

Nous avons introduit le concept de similarité entre les contextes associés aux tâches dans le paragraphe 4.3.1.4, par l'introduction du score de Tversky. Ce score permet de caractériser le pourcentage de similarité entre deux contextes à travers le comptage des éléments communs ou non à ces situations.

Nous proposons dans le présent paragraphe de nous intéresser aux éléments comptés : plus qu'un simple score, il s'agit de regarder quels sont les nouveaux éléments, apportés par l'apparition de la nouvelle tâche, ceux communs aux deux contextes de la transition, et ceux qui ont disparu, n'étant plus considérés comme essentiels à la réalisation de la nouvelle tâche.

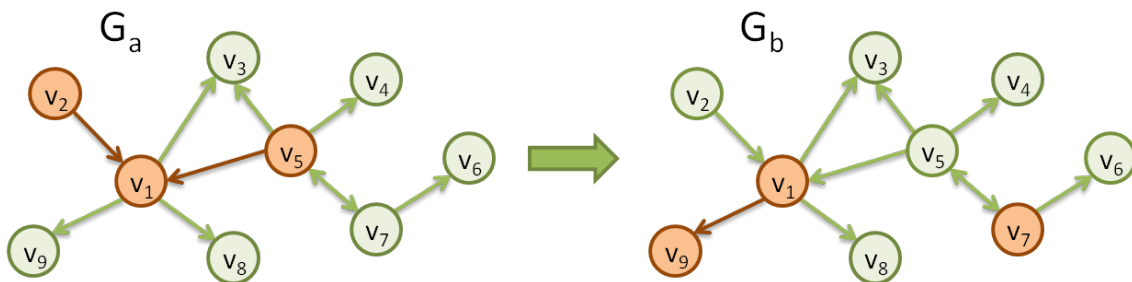


FIGURE 5.12 : un exemple de transition entre deux contextes.

Nouveaux éléments

L'apparition de nouveaux éléments correspond à la définition de nouvelles exi-

gences en termes de *Situation Awareness* pour l'opérateur. Dans la prise en compte de ces éléments pour la conception de l'interface, l'analyste devra tenir compte de la signification donnée pour cet élément vis-à-vis de l'objectif de la tâche à laquelle il est associé. Cet élément peut être nécessaire pour comprendre la situation (niveau 2 de la SA de Endsley (1995b)) ou bien pour anticiper sur une situation future (niveau 3 de la SA de Endsley). De plus, la criticité de l'apparition de cet élément pour la tâche considérée peut être variable. Dans la suite de ce chapitre, nous introduisons des méthodes afin d'étudier la saillance des éléments pour la tâche (section 5.3.4) ou d'ajouter des tags afin de préciser le rôle des éléments dans la SA associée à une tâche (section 5.3.5).

Sur la figure 5.12, les *nouveaux éléments* associés à G_b sont les éléments v_7 et v_9 .

Éléments communs

Il est aussi important d'appuyer la conservation de la SA sur les éléments apportés par la nouvelle tâche que sur les éléments conservés, déjà nécessaires à l'exécution de la tâche précédente. En effet, l'apparition de nouveaux éléments ne doit pas accaparer l'opérateur au risque de perdre la connaissance d'anciens éléments qui seraient pourtant essentiels à la réalisation de la tâche. Les transitions de type « ajout de tâche » illustrent ce cas de figure : la nouvelle tâche devant être exécutée en simultané de la première, il ne faut pas que cela soit effectué au détriment de celle-ci.

Dans la prise en compte de ce principe, il faut considérer bien sûr la priorité des tâches et l'importance des éléments par rapport à ces tâches. Par exemple, dans le cas d'alarmes ou de pannes à gérer, cas typiquement urgent, il faut minimiser l'impact de ces alarmes sur l'activité courante (principe n°31 défini par Endsley *et al.* (2003)).

Sur la figure 5.12, il y a un *élément commun* associé aux contextes G_a et G_b , il s'agit de l'élément v_1 .

Éléments disparus

Les éléments requis pour l'exécution de la tâche précédente, mais non exigés par la nouvelle tâche, représentent un intérêt moindre pour la transition. Néanmoins, dans le cas des interruptions, il peut être intéressant, voire nécessaire de permettre à l'opérateur de garder ces informations disponibles afin de faciliter la reprise de la tâche primaire lorsque l'interruption sera terminée.

De manière générale, il est important de fournir une SA « globale », c'est-à-dire de lui permettre d'avoir accès à un certain nombre d'informations définies par l'analyste, quelque soit la tâche en cours, et donc les exigences particulières en termes de SA pour la réalisation de cette tâche. Selon Endsley *et al.* (2003, principe n°4), cette SA globale est « critique pour permettre à l'opérateur de déterminer de façon précise quels sont les objectifs prioritaires et permettre d'anticiper sur la suite de la mission ».

Sur la figure 5.12, les *éléments disparus* associés à G_a sont les éléments v_2 et v_5 .

5.3.2.2 Lien avec la taxinomie

Nous avons défini dans le paragraphe 5.2.2.5 les correspondances entre les identificateurs CTT et les trois catégories *temporelles* de la taxinomie (introduites dans la section 4.1.2.1), à savoir « transitions simples », « ajout de tâche » et « interruption » (tableau 5.2),

Cette catégorisation temporelle des transitions simples peut avoir un impact en termes de conception d'interface : la prise en compte des éléments d'information nécessaires à la réalisation de chaque tâche ne sera pas les mêmes si la transition est simple ou s'il s'agit d'une interruption ou d'un ajout de tâche.

- Dans le cas des *ajouts de tâche*, la tâche initiale doit toujours être maintenue et donc il faut permettre à l'opérateur de réaliser plusieurs tâches simultanément, sans entraver sa performance.
- Dans le cas des *interruptions*, il faut prendre en compte le fait que, même s'ils ne sont pas forcément nécessaires à la réalisation de la tâche secondaire, les éléments d'information associés à la tâche primaire vont « réapparaître » et il peut être intéressant de faciliter la reprise de cette tâche primaire.

De plus, en considérant les contextes associés aux tâches impliquées dans la transition, et en étudiant particulièrement s'ils possèdent des éléments communs ou sont inclus l'un dans l'autre, nous pouvons également faire correspondre à cette transition une catégorie *contextuelles* de la taxinomie (paragraphe 4.1.2.2).

Ainsi, à partir des catégories temporelle et contextuelle, nous pouvons classer la transition dans la taxinomie (tableau 4.2) et en prédire la difficulté (dont nous discutons dans le paragraphe 4.3.1.3).

5.3.2.3 Analyse de la frontière

Parmi les nouveaux éléments, associés à la nouvelle tâche mais pas à l'ancienne, certains sont directement reliés – par une arête du graphe – à des éléments associés à l'ancienne tâche. Le fait que ces éléments soient situés à la « frontière » entre les deux contextes peut avoir une signification particulière. En effet, dans les réseaux propositionnels, les arêtes entre les nœuds portent un intérêt particulier, de par la méthode de construction de ces graphes (dont les nœuds n'ont pas été ajoutés de manière aléatoire).

Ainsi, dans l'analyse d'une transition, l'analyste pourra prêter une attention particulière aux cas de désactivation d'un élément et de l'activation d'un élément directement adjacent. Ce cas est illustré dans la figure 5.12 par la désactivation de $v_5 \in G_a$ et l'activation de $v_7 \in G_b$.

Ce type de « saut » dans les contextes associés peut être par exemple produit dans le cas où l'opérateur change de référentiel, résultat d'un changement de mode. Nous rencontrons ce cas à plusieurs reprises dans les applications présentées dans

ce mémoire. Par exemple, dans le cas de l'étude du jeu de combat *Battlefield 3* (section 5.5), le joueur lors du passage d'un mode « marche à pied » à un mode de « conduite de tank », va changer de référentiel. Dans le nouveau référentiel, la vue est centrée sur le tank et non plus sur le joueur. Ces éléments sont traduits dans le graphe par le « saut » du nœud « *Player* » associé à la première tâche de *marche à pied* vers le nœud « *tank* », associé à la tâche de *conduite de tank*.

Dans la section suivante, nous proposons d'approfondir la prise en compte des liens entre les nœuds des sous-graphes représentant chaque contexte afin d'obtenir davantage d'informations sur l'impact de la transition.

5.3.3 Recherche de sous-graphes communs

Dans les réseaux propositionnels, la présence ou l'absence d'arêtes orientées entre chaque éléments de situation porte une signification particulière car celles-ci traduisent les liens détectés lors de l'extraction de connaissances entre ces éléments, représentant notamment le cheminement de la réflexion d'un opérateur lors d'un entretien.

Dans la section précédente, nous nous sommes intéressés à la comparaison des éléments communs aux contextes de chaque tâche. Nous souhaitons dans cette section conserver l'information de connexité entre ces éléments et proposons ainsi d'utiliser la **méthode de recherche des plus grands sous-graphes communs**.

Comme son nom l'indique, la méthode de recherche des plus grands sous-graphes communs (*Largest Common SubGraphs* ou LCSG en anglais) a pour but de comparer les deux graphes G_a et G_b , sous-graphes du réseau propositionnel global G dans le but d'en extraire des sous-graphes communs, c'est-à-dire les sous-ensembles composés de nœuds et des arcs reliant ces nœuds, communs à ces deux réseaux propositionnels. Nous présentons dans cette section l'algorithme mis en place pour extraire ces plus grands sous-graphes communs et illustrons ensuite son utilisation sur un exemple.

5.3.3.1 L'algorithme LCSG

Le domaine de la mise en correspondance de graphes (*Graph Matching* en anglais) introduit ce principe de comparaison de graphes pour rechercher leurs similarités. Ce domaine de recherche est particulièrement intéressant pour l'analyse automatisée d'images (Bengoetxea, 2002). Les algorithmes utilisés tentent alors de mettre en correspondance deux graphes indépendants, construits à partir de données issues de deux images, afin d'en détecter les caractéristiques communes (ou non). Un exemple d'utilisation de ce type d'algorithme est la reconnaissance faciale, où de façon simplifiée, les différents éléments d'un visage sont représentés par des nœuds, liés entre eux selon leurs position dans l'image, l'algorithme compare alors l'image qui lui est

soumise à un ensemble d'images de références, sans certitude que tous les nœuds présents dans une image existent dans l'autre.

Nous proposons d'utiliser l'algorithme de recherche des plus grands sous-graphes communs (Shearer *et al.*, 2001) que nous adaptons à notre cas d'usage. En effet, cet algorithme permet initialement de comparer et d'extraire les plus grands sous-graphes communs connexes à deux graphes *indépendants*, c'est-à-dire construits à partir de deux sources (des images dans ce cas) différentes. Dans notre cas d'étude, les contextes G_a et G_b sont tous deux des sous-graphes du graphe global G . Ils sont donc dépendants l'un de l'autre : si deux sommets liés dans G_a sont également présents dans G_b , alors ils seront liés dans G_b .

Ainsi, nous proposons un algorithme simplifié de recherche de ces plus grands sous-graphes communs à $G_a = (V_a, E_a)$ et $G_b = (V_b, E_b)$. Cet algorithme produit un ensemble de sous-graphe de G , tous inclus dans $G_a \cap G_b$. Pour la description de son fonctionnement, nous notons $W = V_a \cap V_b$ et $F = E_a \cap E_b$ et $H = LCSG(G_a, G_b)$ l'ensemble des plus grands sous-graphes communs, résultats de l'algorithme 5.1.

Listing 5.1: algorithme de recherche des plus grands sous-graphes communs.

```

FONCTION LCSG(W)
  H=[]
  Tant que W ≠ ∅
    w = W[0]
    (noeuds, arcs) = TrouverPlusGrandGraphe(w, (W,F))
    Ajouter((noeuds, arcs), H)
    Retirer(noeuds, W)
    Retirer(arcs, F)
  Fin Tant que
  Renvoyer H
FIN FONCTION

```

L'algorithme 5.1 fait appel à une fonction *TrouverPlusGrandGraphe(w, (W,F))* qui permet de rechercher le plus grand graphe dans (W,F) à partir d'un nœud w. Nous la décrivons dans le listing 5.2.

Listing 5.2: algorithme de recherche du plus grand sous-graphe de (W,F) à partir du nœud w.

```

FONCTION TrouverPlusGrandGraphe(w, (W,F))
  temp=copie(W)
  resultat=[]
  aVerifier=[w]
  Tant que aVerifier ≠ ∅
    voisins = TrouverLesVoisins(aVerifier[0])
    Si voisins ≠ ∅

```

```

        Retirer(voisins, temp)
        Ajouter(voisins, aVerifier)
        Ajouter(voisins, resultat)
    Fin Si
    Retirer(aVerifier[0], aVerifier)
Fin Tant que
Renvoyer resultat
FIN FONCTION

```

L'algorithme de recherche du plus grand graphe à partir d'un élément travaille sur un graphe temporaire *temp* et peut être décrit ainsi :

- Tant que la liste des nœuds « à vérifier » est non vide
 - Chercher les voisins du premier nœud de la liste « à vérifier » ;
 - S'il en existe :
 - On supprime ces voisins du graphe *temp* ;
 - On ajoute ces voisins à la liste des nœuds à vérifier ;
 - On ajoute ces voisins à la liste résultat.
 - On retire le nœud dont on vient de chercher les voisins de la liste « à vérifier ».

Notons que l'algorithme travaille uniquement sur l'ensemble des nœuds, l'ensemble des arêtes intervient directement à travers la fonction *TrouverLesVoisins* (non décrite), tous les nœuds appartenant au même graphe global.

A l'issue de l'application de l'algorithme LCSG, nous obtenons un ensemble H des plus grands sous-graphes connexes de W , ces graphes connexes de H ne sont pas connectés entre eux.

5.3.3.2 Illustration

La figure 5.13 illustre l'application de cette méthode d'extraction des plus grands sous-graphes communs à deux contextes

La figure 5.13a présente les contextes initiaux décrits par les sous-graphes G_a et G_b ainsi que l'intersection des ensembles de sommets de ces deux graphes, W .

La figure 5.13b décrit les deux itérations de la boucle de l'algorithme 5.1 : la première itération permet d'extraire le sous-graphe de gauche, à travers 3 itérations de la boucle « Tant que » de l'algorithme d'extraction de plus grand graphe 5.2 (le nœud $w \in W$ arbitrairement choisi est le nœud le plus foncé). Une deuxième occurrence de l'algorithme 5.2 est représentée au centre, permettant de définir un deuxième plus grand graphe (en deux itérations cette fois, notées 4 et 5). Les deux plus grands graphes extraits sont représentés en mauve, sur la droite de la figure 5.13b.

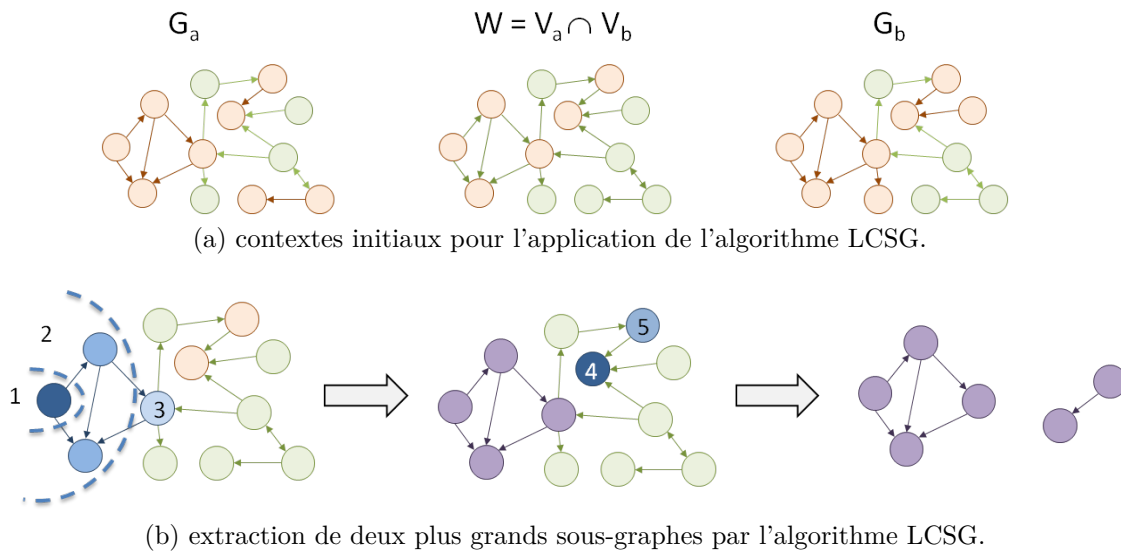


FIGURE 5.13 : illustration de l'application de l'algorithme LCSG.

5.3.3.3 Liens avec l'interface

Les sous-graphes obtenus à travers cette méthode forment des *clusters* ou groupes d'informations. La méthode d'extraction des plus grands sous-graphes communs permet de définir les groupes d'éléments de SA qui sont importants pour la réalisation des deux tâches composant la transition.

Les *clusters* ainsi extraits représentent un ensemble d'éléments de SA liés, et appartenant à un même groupe sémantique. Dans la littérature, il est recommandé de grouper visuellement les informations de même nature et permettant d'atteindre un même but, afin d'éviter à l'opérateur de devoir chercher à travers l'interface (pouvant être composée de plusieurs écrans) ces différentes informations. Ce concept de groupement est notamment porté par le principe n°21 de Endsley *et al.* (2003) ou encore, dans le domaine de l'ergonomie pour la conception d'interfaces, par le critère de *Groupement/Distinction entre Items* de Bastien et Scapin (1993). Par exemple, dans le cas d'un système de pilotage d'un drone, les informations de cap, roulis, tangages et altitude constituent une des bases du pilotage en mode manuel et on pourra les afficher dans la même zone de l'écran.

De plus, afin de garder une certaine consistance entre les tâches pour lesquelles ces informations sont exigées, il est recommandé de fixer ces zones au niveau de l'interface, afin que l'opérateur puisse y accéder rapidement, quelque soit la tâche exécutée (notamment principe n°23 de Endsley *et al.* (2003)). En effet, dans les systèmes complexes, où l'interface peut changer plusieurs fois et rapidement lors du déroulement de la mission, il est important d'avoir ces zones définies où les

informations sont fixées, notamment lors de transition nécessitant de conserver ces *clusters* d'informations en mémoire, mais également afin d'appuyer la conservation d'une SA globale.

5.3.4 Analyse des éléments saillants

En psychologie cognitive, la saillance d'un élément par rapport à d'autres éléments d'un ensemble traduit la démarcation de cet élément par rapport aux autres. Dans le cas des processus perceptifs, la saillance peut être visuelle, l'élément se détache alors par son aspect différent des autres (un carré bleu dans un groupe de cercles rouges par exemple). La saillance visuelle est d'ailleurs l'une des composante du modèle SEEV (*Salience, Expectancy, Effort, Value*) de Wickens *et al.* (2003), intégré au module de *Situation Awareness* de l'outil MIDAS (§ 3.2.2.1).

Dans notre cas, nous cherchons à définir quels sont les éléments saillants *a priori* pour l'opérateur, c'est-à-dire quels sont les éléments de SA les plus pertinents parmi l'ensemble des exigences définies pour une tâche donnée. A cette fin, nous étudions deux méthodes de définition de la saillance : une méthode quantitative, proposée par Stanton *et al.* (2006) et une méthode plus qualitative, que nous introduisons.

5.3.4.1 Calcul de la saillance selon la méthode de Stanton et al.

Lors de l'introduction du modèle DSA, Stanton *et al.* (2006) définissent les éléments de situations les plus *saillants* comme « les objets de connaissance qui exercent un rôle central par rapport aux autres objets de connaissance (c'est-à-dire qui ont cinq arêtes ou plus les reliant à d'autres objets de connaissance) »⁴.

Ce calcul peut être mené sur l'ensemble du graphe global afin d'en extraire les éléments saillants, indépendamment de la tâche considérée. Ensuite, pour chaque tâche impliquée dans une transition, l'analyste pourra regarder quels sont les éléments saillants faisant partie des exigences en termes de *Situation Awareness* définies pour les tâches étudiées et ainsi définir quels sont les éléments de situation les plus important avant et après la transition.

Sur la figure 5.14, le nœud v_1 est saillant au sens de Stanton pour la tâche $T_{1.2}$.

5.3.4.2 Définition qualitative de la saillance

En parallèle de la définition quantitative proposée par Stanton et ses collègues, nous souhaitons donner à l'analyste la possibilité de définir précisément, de manière qualitative, quels sont les éléments les plus pertinents pour chaque tâche. En effet, en appliquant la méthodologie à un cas opérationnel (chapitre 7), nous nous sommes aperçu que certains des éléments de SA jugés très importants pour la réalisation

4. en anglais, « *those knowledge objects that serve as a central hub to other knowledge objects (i.e., have five or more links to other knowledge objects)* » (Stanton *et al.*, 2006)

de certaines tâches n'étaient pas nécessairement des éléments centraux (au sens de Stanton) dans le graphe global mais pouvaient avoir un impact sur la SA lors d'une transition.

Ainsi, cette deuxième définition de la saillance pourra être utilisée par l'analyste afin de traduire dans le modèle, lors de l'étape d'association, la pertinence exprimée par les opérateurs (ou déduite de la littérature) pour certains éléments. Afin de visualiser les éléments saillants du sous-graphe définissant un contexte, l'analyste pourra choisir de les faire apparaître dans une couleur différente de celle utilisée pour visualiser ces sous-graphes, par exemple une teinte plus foncée de la couleur utilisée pour visualiser ces sous-graphes, par exemple une teinte plus foncée de la couleur utilisée permettra de garder le lien avec le sous-graphe tout en renforçant l'importance de ces éléments plus foncés.

Nous avons appliqué ce principe sur la figure 5.14, où le nœud v_5 est défini comme saillant pour la tâche $T_{1.2}$ par l'analyste.

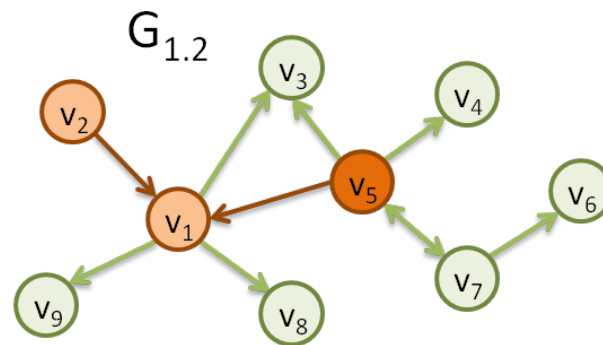


FIGURE 5.14 : visualisation de la saillance d'un élément (nœud v_5) définie qualitativement.

5.3.4.3 La saillance dans les transitions

La prise en compte de ce concept de saillance dans les interfaces peut-être effectuée à deux niveaux :

- au niveau des tâches seules, à travers la mise en exergue de ces éléments d'information afin de permettre la réalisation de ces tâches. Par exemple, en reprenant le principe de saillance visuelle, ces éléments peuvent être mis en avant en différenciant la forme, la couleur, le type de la représentation de cette information sur l'interface.
- au niveau des transitions, afin de faciliter l'activation de la nouvelle tâche. C'est le principe n°6 de Endsley *et al.* (2003) : « *Make critical cues for schema activation salient* »⁵. De la même façon, on s'appuiera sur la modalité visuelle pour afficher les changements d'états des éléments saillants lors de la transition, et notamment mettre en exergue les nouveaux éléments, associés à la

5. « Rendre saillantes les indications critiques à l'activation de schémas »

nouvelle tâche.

Endsley *et al.* (2003) indiquent également qu'il peut être intéressant d'utiliser la saillance visuelle afin d'appuyer la mise à jour des informations affichées (principe n°11), que ce soit en phase de transition mais également en fonctionnement « normal ». De la même façon, lors de changement de modes (traduisant différents niveaux d'autonomie) ou d'activation / désactivation de systèmes particuliers, il est nécessaire de rendre saillant le nouvel état de l'élément considéré afin d'éviter toute confusion (principe n°39).

Par exemple, dans le crash du McDonnell Douglas MD-82 d'American Airlines (vol 1420) ayant causé le décès du capitaine et de dix passagers, une des erreurs mise en avant dans le rapport d'accident (NTSB, 1999) est l'oubli de la part des pilotes d'embrayer les aérofreins (*spoilers*) en mode automatique. La figure 5.15 permet d'observer la différence entre le cas de figure où les aérofreins ne sont pas armés (à gauche) et celui où ils sont armés (à droite). La différence est visible par une partie rouge supplémentaire, sous le levier correspondant, dont la saillance visuelle pourrait être améliorée (ou redondée).



FIGURE 5.15 : différence sur le McDonnell Douglas MD-82 entre un mode où les aérofreins ne sont pas armés (à gauche) et armés (à droite) (extrait de NTSB, 1999).

Dans le cas des interruptions de tâches, nous avons noté dans notre état de l'art (section 2.3.2.1) que les indications visuelles facilitent la reprise de la tâche primaire en indiquant notamment l'état du système avant l'interruption afin de rappeler les dernières actions effectuées, permettant de faciliter la récupération de la SA sur la tâche primaire.

5.3.5 Analyse des métadonnées associées à la transition

Dans certains cas, il peut être pertinent d'ajouter des informations supplémentaires aux éléments de situations décrits dans le réseau propositionnel. En effet, la représentation des exigences de SA sous forme de réseau propositionnel implique de

restreindre la définition de ces éléments à un nombre de mots réduit. Ceci présente l'avantage d'éviter les écueils des descriptions trop prolixes, et permet ainsi de focaliser le réseau sur le concept en jeu et non sur sa description (au risque d'aller jusqu'à la description de solutions techniques). Néanmoins, selon les cas, il peut être intéressant de pouvoir ajouter certaines informations, sous forme de méta-données, afin de caractériser un élément de situation.

Nous proposons ainsi d'introduire une possibilité de taguer ces éléments avec une ou plusieurs étiquettes, traduisant certaines informations devant être prises en compte lors de la conception. Ces données seront de nature descriptive et qualitative, et propres à chaque tâche (de la même façon que nous avons introduit la saillance définie par l'utilisateur sous la forme d'un booléen dans la sous-section précédente).

A travers ces étiquettes, l'analyste pourra ajouter des informations sur l'aspect dynamique des éléments de situation, la représentation typique de ces informations dans le cas de l'exécution de la tâche considérée. L'ajout de ces méta-données pourra également traduire des précisions apportées par les experts lors des entretiens, ou par les auteurs de rapports et autres travaux sur le domaine, afin d'intégrer ces détails pertinents. Enfin, ces étiquettes pourront permettre de spécifier quels aspects de la SA, au sens de Endsley (compréhension / projection), doivent être appuyés par l'interface pour cet élément.

Il faut cependant s'assurer que ces tags ne soient pas redondants avec le contenu du réseau propositionnel et ne pas créer, autour d'un élément particulier, une sorte de « sous-réseau d'informations » sous la forme de tags associés à cet élément. On veillera ainsi à limiter le nombre de ces tags à des concepts de description de l'information étiquetée.

Dans le cas de l'analyse des transitions, ces étiquettes peuvent impacter la représentation des informations nécessaires à l'exécution des tâches de différentes façons :

- si un élément associé à la nouvelle tâche est tagué pour cette tâche particulièrement, il faut prendre en compte les données contenues dans ce tag ;
- si un élément est commun aux deux tâches impliquées dans la transition, et que les tags associés à cet élément pour les deux tâches considérées sont différents, cela peut avoir un impact sur la représentation de l'élément sur l'interface.

Par exemple, lors d'un entretien, un opérateur pourra exprimer, dans le cas d'une tâche de pilotage d'un drone, l'affichage de la vitesse sous la forme d'un vecteur vitesse (représentation graphique, la taille du vecteur varie en fonction de la vitesse, et sa direction correspond au cap du drone) tandis que, lorsqu'il s'occupe de la charge utile (la caméra), il n'a pas besoin de voir le vecteur vitesse mais préférera voir la « trace au sol » de la caméra (la zone couverte pas le champ de la caméra). On conservera alors uniquement l'affichage de la valeur numérique (permettant de connaître la vitesse par rapport au sol de l'aéronef, et ainsi anticiper le contrôle de la caméra).

L'analyste pourra traduire ces commentaires en étiquetant différemment le nœud « vitesse » du réseau propositionnel en fonction de la tâche considérée. Ainsi, lors de la transition de la tâche de pilotage à la tâche de contrôle de la charge utile, nous pouvons imaginer que ce changement de représentation soit appuyé par une animation sur l'interface (voir aussi la partie 5.4.2.2 au sujet des animations). La figure 5.16 illustre une possibilité de transition mettant en avant le changement de visualisation de la valeur de la vitesse du drone, qui est transférée dans une autre zone de l'interface où sont rassemblés les paramètres relatifs à ce vol. Cette transition introduit une étape intermédiaire, où la vitesse est affichée sous le drone au format textuel, avant d'être amenée vers la zone de l'écran où elle est affichée habituellement (l'affichage au format texte est constant, afin de permettre à l'opérateur de conserver une SA globale sur ce paramètre de vol important).

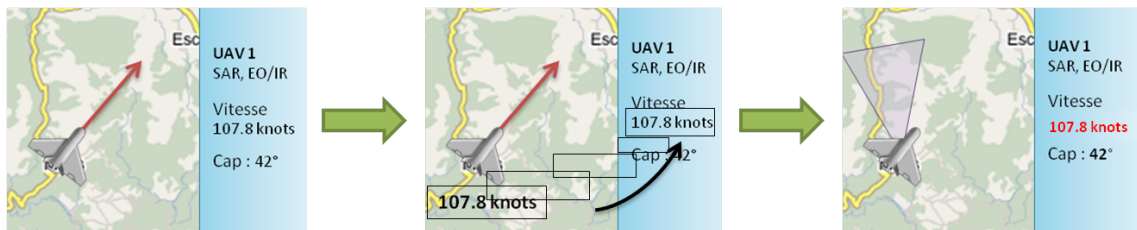


FIGURE 5.16 : exemple de transition visuelle, animation du paramètre de vitesse.

5.3.6 Intégration des tâches parallèles dans l'analyse

L'ensemble des méthodes présentées dans cette section se focalise sur l'analyse des transitions définies par le passage d'une tâche à une autre. Néanmoins, comme nous l'avons exprimé à plusieurs reprises, la complexité des scénarios opérationnels nécessite généralement l'exécution de plusieurs tâches en parallèle des tâches principales.

Par exemple, il peut s'agir de tâches de surveillances de paramètres ou de tâches correspondant à l'orientation dans l'espace de la mission (dans le cas du pilotage, ces tâches représentent des tâches à part entière). Ces tâches « de fond » n'interviennent pas nécessairement dans la tâche courante ou lors de la transition, mais certains des éléments de SA nécessaires à la réalisation de ces tâches complémentaires peuvent influencer la transition et il serait intéressant de les intégrer dans l'analyse.

Lors de la conception de l'interface, le but est de permettre la réalisation de ces tâches en apportant les informations nécessaires à leur exécution, sans toutefois entraver l'exécution de la tâche « primaire ». Ainsi, l'analyste devra garder à l'esprit l'existence de ces tâches « secondaires », notamment lors de l'analyse des transitions. Il pourrait être intéressant d'intégrer, lorsque cela paraît pertinent, ces tâches parallèles à l'analyse afin par exemple de détecter si certains éléments de ces tâches

de fond sont communs à l'une des tâches composant la transition, permettant ainsi de renforcer leur influence sur la conception de l'interface.

5.3.6.1 Recherche des tâches parallèles

Lors de l'analyse, les tâches pouvant être exécutées en parallèle de l'une ou l'autre tâche impliquée dans la transition sont accessibles directement dans l'arbre CTT à travers l'identificateur de concurrence (avec ou sans transmission d'information). La présence de cet identificateur dans l'arbre CTT ne signifie pas que les tâches sont nécessairement exécutées en parallèle mais que leurs intervalles d'exécution se superposent, comme le précise la correspondance avec les relations d'intervalle de Allen présentée dans le tableau 5.2.

Pour une tâche particulière, l'algorithme de recherche des tâches pouvant s'exécuter en parallèle consiste à prendre l'ensemble des ascendants de cette tâche et vérifier, pour chacun, si cet ascendant possède des tâches parallèles. La figure 5.17 présente sous forme graphique la mise en œuvre de cet algorithme :

- la tâche $T_{2.3.1}$ possède une tâche parallèle, la tâche $T_{2.3.2}$;
- elle a également deux ascendants (sans compter la racine), les tâches $T_{2.1}$ et T_1 . Parmi ses ascendants, nous trouvons une deuxième tâche pouvant être exécutée en parallèle, la tâche T_1 .

Ainsi, les tâches parallèles à la tâche $T_{2.3.1}$ sont les tâches $T_{2.3.2}$ et T_1 .

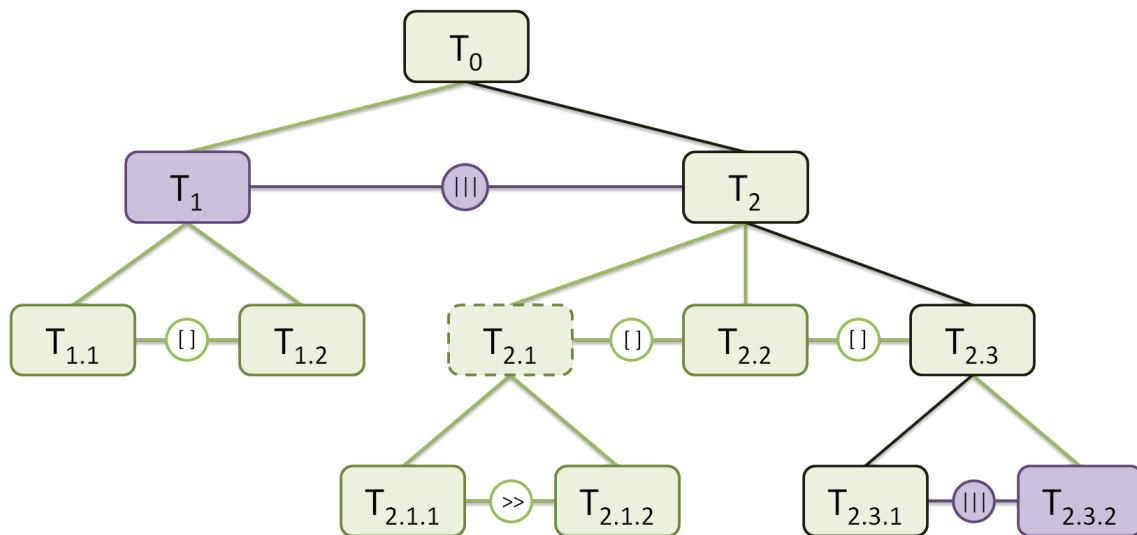


FIGURE 5.17 : recherche des tâches parallèles à la tâche $T_{2.3.1}$, ses ascendants sont forcés, les tâches parallèles sont colorées en mauve.

5.3.6.2 Mise en œuvre de l'algorithme pour l'analyse des transitions

Lors de l'analyse d'une transition, l'analyste pourra ainsi rechercher les tâches parallèles à la transition. Ces tâches sont celles pouvant être exécutées en parallèle des deux tâches associées à la transition.

Par exemple, sur la figure 5.18, nous étudions la transition de la tâche $T_{2.1.2}$ vers la tâche $T_{2.3.1}$. La tâche T_1 peut être exécutée en « tâche » de fond pendant toute la durée de la transition, et pendant l'exécution subséquente de la nouvelle tâche $T_{2.3.1}$.

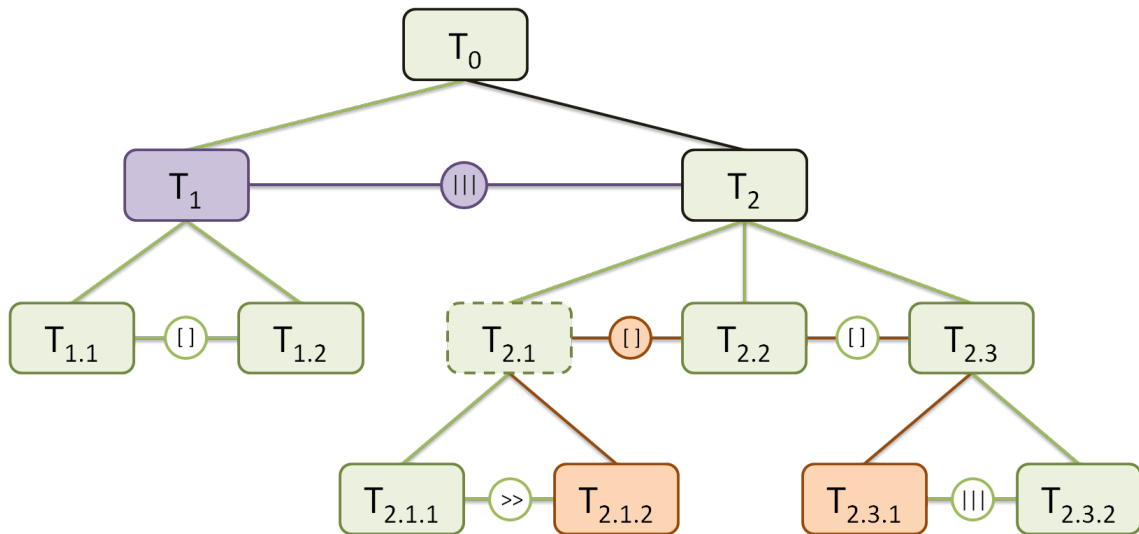


FIGURE 5.18 : recherche des tâches exécutées en parallèle d'une transition, la tâche T_1 est présente pendant toute la transition de la tâche $T_{2.1.2}$ vers la tâche $T_{2.3.1}$.

5.3.6.3 Discussion

La recherche des tâches parallèles dans le cadre d'une analyse des transitions permet d'ajouter un complément d'information sur la transition considérée. Néanmoins, tous les cas de figure ne sont pas intéressants et l'ajout des tâches parallèles à l'analyse peut venir « polluer » l'analyse initiale de la transition.

En effet, lors de l'intégration de ces tâches à l'analyse, nous avons utilisé l'opération d'union ensembliste (tableau 5.3) pour intégrer le contexte associé à cette tâche parallèle à l'analyse. Lorsque ce contexte apporte beaucoup d'éléments, l'ajout de ces éléments à ceux des contextes de tâches impliquées dans la transition peut venir « fausser » l'analyse, notamment l'extraction des sous-graphes communs. Nous préférons donc utiliser cette intégration des tâches parallèles au cas par cas, et laissons ainsi à l'analyste le soin de juger de la pertinence d'intégrer l'une ou l'autre de ces tâches.

Cette prise en compte des tâches parallèles peut également se faire de manière spécifique à l'une ou l'autre des tâches impliquées dans la transition. Par exemple, dans notre illustration (figure 5.18), la tâche $T_{2.3.2}$ ne peut être exécutée qu'en parallèle de la tâche $T_{2.3.1}$, qui est la nouvelle tâche de la transition. Le contexte associé à cette tâche est donc pertinent uniquement lors de l'étude des « nouveaux éléments » et peut être pris en compte par l'analyste dans ce cas de figure.

Enfin, les règles de « factorisation » que nous avons introduites afin de lever les ambiguïtés (§ 5.2.2.2) présentent certaines limites : elles empêchent la prise en compte des tâches « filles » de la tâche parallèle considéré. Sur la figure 5.18, la tâche T_1 exécutée en parallèle de la transition considérée regroupe deux sous-tâches, liées par un identificateur d'alternative, indiquant que l'une ou l'autre de ces tâches est exécutée. Dans certains cas, il peut être davantage pertinent de s'intéresser au contexte de l'une de ces sous-tâches ($T_{1.1}$ ou $T_{1.2}$) qu'à celui, probablement réduit, associé à la tâche « mère » parallèle (T_1).

Ce cas de figure est par exemple rencontré lorsqu'un opérateur a la possibilité de piloter un système selon plusieurs modes, à choisir entre un mode manuel ($T_{1.1}$) ou un mode de contrôle automatique ($T_{1.2}$), tous les deux regroupés sous une tâche « mère » de pilotage (T_1).

Dans ces cas de figure, les éléments associés au mode de pilotage peuvent être pertinents pour la transition étudiée, mais l'application stricte de notre algorithme ne permet pas d'accéder aux tâches propres à chaque mode, mais uniquement à la tâche « mère ». Afin de contourner cette limite de la « factorisation des tâches », nous proposons de ne pas considérer la tâche « mère » mais l'une des tâches « filles » en tant que tâche « de fond » pour l'analyse, en maintenant cette tâche tout au long de l'analyse de la transition. Cette tâche « adhésive » précisera le contexte de l'analyse. Nous rencontrons ce cas de figure dans l'exemple applicatif présenté au chapitre 7, portant sur les systèmes de drone, où la tâche de pilotage est une tâche « de fond » lorsque l'opérateur du drone se focalise sur la gestion de sa caméra.

L'intégration des tâches parallèles dans une analyse des transitions peut augmenter rapidement la complexité de cette analyse, en particulier si cela est fait systématiquement. Le jugement s'effectuera au cas par cas, en fonction de la pertinence des tâches parallèles considérées et de leur apport potentiel au contexte de la transition étudiée.

5.4 IMPACT DES TRANSITIONS SUR L'IHM (PHASE 2, ÉTAPE 5)

La dernière étape de la méthodologie est très liée à l'étape précédente d'analyse des transitions, puisqu'elle consiste à traduire les résultats de l'analyse en recommandations pour la conception des interfaces. Ces recommandations peuvent être ensuite traduites sous forme de maquettes ou de prototypes d'interfaces, et validées à travers des tests utilisateurs.

Nos travaux ne se focalisant pas sur cette dernière étape, nous proposons cependant quelques sources sur lesquelles peuvent s'appuyer analystes et concepteurs pour la conception d'interfaces des systèmes analysés. Dans un deuxième temps, nous ferons le lien entre ces sources et les modèles et méthodes d'analyse présentés dans la description de l'étape 4.

5.4.1 Les normes et guides en ergonomie et facteurs humains

Afin de produire des recommandations pertinentes suite aux analyses des transitions, l'analyste, pourra s'appuyer sur les normes, guides et critères existants dans le domaine de l'ergonomie et des facteurs humains. Nous en présentons quelques-unes dans cette sous-section.

5.4.1.1 Les principes de conception intégrant la Situation Awareness

Comme nous l'avons présenté au chapitre 3, la littérature sur la SA est vaste et ce concept répandu, la majorité des guides et normes du domaine des facteurs humains comporte a minima une section portant sur l'intégration de la SA dans la conception.

Au cours de la rédaction de ce chapitre, et dans le cadre de cette thèse, nous nous sommes appuyé en particulier sur l'ouvrage de Endsley *et al.* (2003) intitulé « *Designing for Situation Awareness* ». Ce livre aborde la SA sous différents angles (intégration des alarmes, de l'automatisation, de la complexité des systèmes) et propose 50 principes généraux, que nous reprenons dans les tableaux 5.4 et 5.5.

Bien que liés au modèle linéaire à trois niveaux de la SA (perception, compréhension, projection), ces principes demeurent applicables à notre cas de figure. En particulier, certains des principes sont directement liés aux modèles et concepts mis en avant dans la méthodologie SAMANTA. Par exemple, le principe n°1 « Organiser les informations autour des buts » (tableau 5.4) peut être directement lié à l'arbre de tâches, dont la construction est basée sur la définition de buts et sous-buts. Le principe n°6 « Rendre saillantes les indications critiques pour l'activation de schéma » (tableau 5.4) peut être appliqué aux *nouveaux éléments* apparaissant dans une transition, et liés à un nouveau schéma du graphe de situation associé, en particulier quand ces éléments sont saillants dans le graphe. Faire en sorte que ces nouveaux éléments soient mis en exergue lors de la transition peut aider l'opérateur à démarrer plus rapidement sa nouvelle tâche.

5.4.1.2 Les normes et guides spécifiques aux systèmes complexes

Nous présentons dans ce paragraphe trois références portant sur l'intégration des facteurs humains dans la conception de systèmes complexes, et particulièrement pour la conception des interfaces. Ces documents ne sont pas spécifiquement « orientés

CHAPITRE 5. LA MÉTHODOLOGIE SAMANTA

TABEAU 5.4 : résumé des principes de conception intégrant la SA, première partie
(extrait de Endsley *et al.*, 2003)

No.	SA Design Principles	Page
General		
1	Organize information around goals	83
2	Present Level 2 information directly—support comprehension	83
3	Provide assistance for Level 3 SA projections	85
4	Support global SA	86
5	Support trade-offs between goal-driven and data-driven processing	86
6	Make critical cues for schema activation salient	86
7	Take advantage of parallel processing capabilities	87
8	Use information filtering carefully	87
Certainty Design Principles		
9	Explicitly identify missing information	123
10	Support sensor reliability assessment	125
11	Use data salience in support of certainty	126
12	Represent information timeliness	128
13	Support assessment of confidence in composite data	129
14	Support uncertainty management activities	130
Complexity Design Principles		
15	Just say no to feature creep—buck the trend	144
16	Manage rampant featurism through prioritization and flexibility	144
17	Insure logical consistency across modes and features	145
18	Minimize logic branches	145
19	Map system functions to the goals and mental models of users	145
20	Provide system transparency and observability	146
21	Group information based on Level 2/3 SA requirements and goals	146
22	Reduce display density, but don't sacrifice coherence	146
23	Provide consistency and standardization on controls across different displays and systems	147
24	Minimize task complexity	147
Alarm Design Principles		
25	Don't make people rely on alarms—provide projection support	164
26	Support alarm confirmation activities	167
27	Make alarms unambiguous	167
28	Reduce false alarms, reduce false alarms, reduce false alarms	167
29	Set missed alarm and false alarm trade-offs appropriately	168
30	Use multiple modalities to alarm, but insure they are consistent	169
31	Minimize alarm disruptions to ongoing activities	169
32	Support the assessment and diagnosis of multiple alarms	170

TABLEAU 5.5 : résumé des principes de conception intégrant la SA, deuxième partie (extrait de Endsley *et al.*, 2003)

<i>No.</i>	<i>SA Design Principles</i>	<i>Page</i>
33	Support the rapid development of global SA of systems in an alarm state	170
<i>Automation Design Principles</i>		
34	Automate only if necessary	186
35	Use automation for assistance in carrying out routine actions rather than higher level cognitive tasks	186
36	Provide SA support rather than decisions	186
37	Keep the operator in control and in the loop	188
38	Avoid the proliferation of automation modes	188
39	Make modes and system states salient	188
40	Enforce automation consistency	189
41	Avoid advanced queuing of tasks	189
42	Avoid the use of information cueing	190
43	Use methods of decision support that create human/system symbiosis	190
44	Provide automation transparency	190
<i>Multioperator Design Principles</i>		
45	Build a common picture to support team operations	213
46	Avoid display overload in shared displays	215
47	Provide flexibility to support shared SA across functions	215
48	Support transmission of different comprehensions and projections across teams	216
49	Limit nonstandardization of display coding techniques	217
50	Support transmission of SA within positions by making status of elements and states overt	218

Situation Awareness » mais représentent des guides (voire une norme dans le cas du dernier document) pour la conception.

NASA – Human Integration Design Handbook

La NASA (*National Aeronautics and Space Administration*) a publié en 2010 un guide d'intégration des humains dans la conception, « *Human Integration Design Handbook* » (HIDH) qui décrit à travers 9 chapitres les différents aspects de l'intégration du couple humain / système pour le programme de vol habité de la NASA (2010). Il est clairement destiné aux professionnels des facteurs humains en charge de la conception de ces systèmes habités, comme l'indique l'introduction au guide.

Le chapitre 10 correspond plus particulièrement aux exigences vis-à-vis de la conception des interfaces (en termes d'agencement et forme des contrôles, d'affichage, de libellés, de sortie audio et d'interaction homme-machine). Cette section présente à la fois des exigences portant sur l'aspect matériel des contrôles (entrées de l'utilisateur) et des écrans ainsi que des recommandations portant sur la conception des interfaces et des interactions à travers ces interfaces.

FAA – Human Factors Design Guide

Le document « *Human Factors Design Guide For Acquisition of Commercial-Off-The-Shelf Subsystems, Non-Developmental Items, and Developmental Systems* », publié par la FAA (*Federal Aviation Administration*) a été établi afin de fournir aux responsables facteurs humains de la FAA une référence en termes d'exigences facteurs humains au sein de l'organisation. Dans ce document de 1000 pages, le chapitre 8 « *Human-Computer Interfaces* » présente les dispositions relatives aux IHM informatiques. Nous y retrouvons un contenu proche de celui du HIDH de la NASA en termes de composants abordés. D'autres chapitres peuvent être intéressants dans le cadre de l'intégration de la *Situation Awareness* dans la conception, notamment le chapitre 5 portant sur les affichages et le chapitre 7 portant sur les alarmes visuelles et auditives.

Les dispositions sont présentées sous forme de recommandations sur les composants, proposant des exceptions, des exemples de cas d'utilisation et des descriptions pour certains points plus critiques. Les recommandations présentées sont toutes justifiées par des sources (normes, standards ou littérature du domaine facteurs humains) dans l'annexe B du document (FAA, 1996).

DoD - Design Criteria Standard

Le document « *Design Criteria Standard* », référencé MIL-STD-1472G (il s'agit de la 7^{ème} version de cette norme publiée en janvier 2012), et publié par le département de la défense américain DoD (2012), est un standard décrivant les exigences en termes de conception de poste de travail (matériel et interfaces) de façon très précise, en indiquant par exemple, dans le cas des interfaces, les tailles de caractères minimum (en mrad). L'introduction de ce document précise que la mise en application de

ces standards doit se faire sous la directive de spécialistes en facteurs humains qui sauront trouver des solutions aux cas particuliers non couverts par le document.

5.4.1.3 Les critères et norme de conception en ergonomie

La norme ISO 9241

La norme ISO 9241 est en fait un ensemble de documents, regroupés sous l'intitulé « Ergonomie de l'interaction homme-système »⁶ et publiés dans 8 séries (regroupées en centaines, de 100 à 900) portant sur les différents aspects de l'ergonomie pour l'interaction homme-machine : ergonomie logicielle (série 100), processus d'interaction homme-système (série 200), affichage et matériel relatif à l'affichage (série 300), matériel physique d'entrée (série 400), ergonomie du lieu de travail (série 500), ergonomie de l'environnement (série 600), domaines d'application - salles de contrôle (série 700) et interactions tactiles et haptiques (série 900).

Ces normes peuvent être utilisées en phase de conception, afin de guider la conception des interfaces, de manière globale ou en abordant des points plus précis (salles de contrôles, interaction spécifiques tactiles et haptiques) mais peuvent également donner lieu à la mise en place de liste de vérification, afin d'évaluer la conformité d'un système à la norme. Ce processus de vérification peut mener à la certification d'un système (certification ISO 9241).

Les critères de Bastien et Scapin

Bastien et Scapin (1993) ont publié une liste de 18 critères élémentaires, distribués dans 8 catégories, permettant d'évaluer l'aspect ergonomique des interfaces utilisateurs. Les huit catégories présentées sont le guidage, la charge de travail, la contrôle explicite, l'adaptabilité, la gestion des erreurs, l'homogénéité/cohérence, la signifiante des codes et dénominations et enfin, la compatibilité. Chaque critère est défini précisément, justifié, illustré par des exemples de recommandations issues de ces critères et enfin commenté.

Ces critères peuvent être utilisés à la fois en phase de conception, afin de guider le concepteur dans la mise en place de recommandations, et en phase d'évaluation, afin de vérifier qu'un système considéré répond à l'ensemble des critères définis.

5.4.1.4 Conception d'IHM « grand public »

Dans le domaine plus grand public de la conception d'interfaces logicielles ou Web, il existe de nombreux guides, ouvrages ou publications scientifiques portant sur l'ergonomie de ces interfaces.

6. Anciennement intitulée « Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) », cette norme a été renommée en 2006 et sa numérotation revue.

Les guides spécifiques aux systèmes d'exploitation

Les trois systèmes d'exploitation les plus répandus, à savoir Microsoft Windows, Apple Mac OS X et les environnements de bureau de la famille Unix (GNOME et KDE notamment) possèdent tous leur guides de conception d'interface.

Ainsi, le document « *Windows User Experience Interaction Guidelines for Windows 7 & Windows Vista* » présente les différentes recommandations portant sur les interfaces des applications destinées à être exécutées sur les systèmes d'exploitation de la famille Windows (en version Vista & 7). Ce guide très détaillé (882 pages) décrit chaque composant de l'interface Windows ainsi que ses usages recommandés, illustrés d'exemples et contre-exemples. De la même façon, le document « *Mac OS X Human Interface Guidelines* » détaille de façon très précise le comportement ainsi que l'aspect visuel que doit avoir toute application exécutant dans l'environnement MacOS X, mélangeant les principes et recommandations portant sur le graphisme et l'ergonomie. Enfin, pour l'environnement GNOME, on trouvera le « Guide de l'interface utilisateur (GIU) pour GNOME », en version 2.2.2.

Ces guides permettent de définir les « bonnes pratiques » pour la conception d'interfaces dans les environnements considérés mais sont destinés à la conception d'applications et d'interfaces au public plus large que les systèmes complexes considérés dans ce mémoire. Notamment, le concept de *Situation Awareness* n'y est pas traité. Ils sont cependant intéressants à prendre en compte lors de la phase finale de conception d'un système complexe afin de s'assurer de l'homogénéité des comportements interactifs et graphiques vis-à-vis du système d'exploitation utilisé. L'utilisation de paradigme d'interactions classiques et connus permet notamment aux opérateurs de se focaliser sur les tâches complexes, et non sur la compréhension du fonctionnement de l'interface.

La conception d'interface pour le grand public

Le domaine de l'ergonomie étant un domaine en large essor, on trouve de plus en plus d'ouvrages portant sur ce sujet. L'ouvrage de Nogier *et al.* (2011), dans sa cinquième édition, reprend notamment les principes et critères de différents auteurs (critères de Bastien et Scapin (1993) par exemple) afin de guider la conception d'interfaces logicielles ou Web.

5.4.2 Intégration des normes et guides à la méthodologie

5.4.2.1 Lien avec les méta-données

Dans le paragraphe 5.3.5, nous avons introduit le concept de métadonnées pouvant être associées aux nœuds du graphe lors de la définition des contextes.

Ces tags caractérisant les aspects d'affichage peuvent être liés aux différents critères ergonomiques et aux principes énoncés dans la littérature. Imaginons que

chaque nœud soit tagué par une métadonnée indiquant sa modalité de représentation préférée (celle utilisée par défaut sur l'interface), par exemple « analogique » (traduite sous forme de jauge, de flèche, ...) ou « numérique » (plutôt traduite par l'affichage de nombres). Nous pouvons alors appliquer le principe de cohérence de l'interface (Bastien et Scapin, 1993) en le traduisant par une contrainte sur les tags lors d'une transition : si l'ensemble de nœuds partagés lors de la transition est représenté sous forme « analogique » sauf un ou deux, nous pouvons contraindre l'homogénéisation des représentations en imposant aux nœuds « numériques » d'être affichés en analogique, afin d'assurer une consistance de l'interface globale.

Ainsi, ce principe pourrait être généralisé en énonçant que lorsqu'un ensemble de nœuds liés est tagué avec un même tag, alors les nœuds ayant comme voisins une majorité de nœuds « bien tagués » héritent automatiquement de ce tag, lorsqu'il s'agit d'un tag décrivant la modalité de présentation d'une information (figure 5.19).

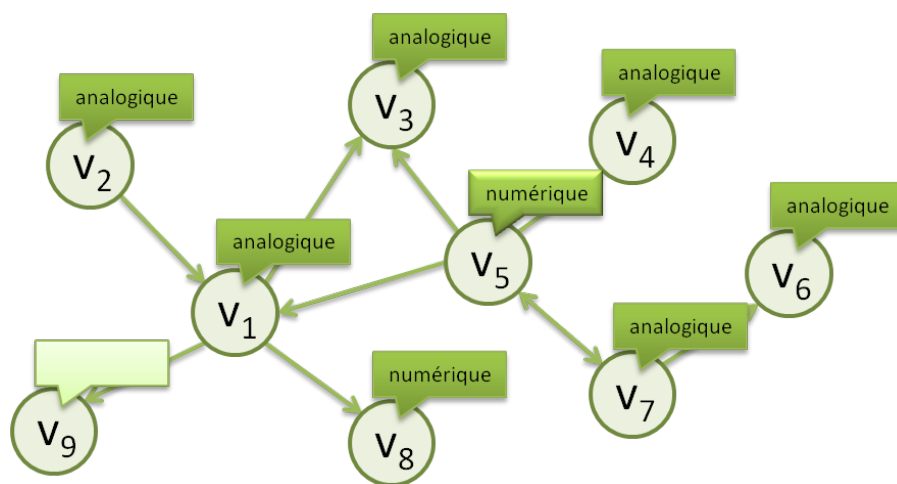


FIGURE 5.19 : utilisation des tags afin d'assurer la cohérence de l'interface. Le nœud v_5 , tagué « numérique » est entouré de nœuds tagués « analogique ». Afin d'harmoniser les représentations, il héritera du tag « analogique ». De même, le nœud v_9 , initialement non tagué, pourra hériter du même tag.

Sur le même principe, dans le cas de la conception d'un système complexe basé sur des normes spécifiques, nous pouvons imaginer de traduire certains des principes ou recommandations émis sous forme de tags pour les éléments. Par exemple, certaines données doivent être codées en couleur dans le cas d'alarmes, en suivant des normes de couleurs spécifiques selon l'importance de l'information pour la tâche effectuée. De la même façon, certaines normes aéronautiques ou militaires imposent la représentation d'informations sous forme de jauges, ou de compteur : cette donnée peut apparaître sous forme de tag pour le nœud considéré.

5.4.2.2 Les animations visuelles pour faciliter les transitions

Afin d'appuyer les transitions, l'analyste peut proposer de mettre en place des animations spécifiques, permettant à l'opérateur de mieux appréhender les changements d'états du système. De nombreux travaux dans la littérature portent sur l'intégration des animations afin d'améliorer la prise de décision des opérateurs. Gonzalez (1996) montre par exemple que la qualité de la décision dépend notamment du réalisme et de la fluidité de l'animation soutenant la transition, et qu'elle est meilleure lorsque la transition est effectivement animée et non instantanée.

L'usage des animations visuelles pour soutenir les transitions dans les interfaces utilisateurs peut être traduit de multiple façons. Nous n'en ferons pas un état de l'art complet ici : Dessart *et al.* (2011), en introduction d'un article sur l'adaptabilité des interfaces, présentent une liste d'applications possibles, comme l'animation des retours visuels des actions utilisateurs, les notifications, la mise en avant de liens causaux ou spatiaux, etc.

Une approche intéressante pour la prise en compte des transitions dans l'animation est celle de Schlienger *et al.* (2006), qui introduisent le concept de « transition visuelle différenciée », permettant de faire varier l'animation associée à une transition en fonction des paramètres des concepts manipulés. Par exemple, la visualisation du transfert d'un fichier « lourd » est traduite par une animation plus lente que pour le transfert d'un fichier de petite taille, reproduisant l'effet d'une pseudo-gravité dépendant de la taille du fichier.

Cette idée de transition s'appuie sur le paramétrage des variables modifiées lors de la transition. Nous pouvons adapter ce principe aux transitions analysées à travers la méthodologie SAMANTA, en s'appuyant notamment sur la notion « d'éléments frontières ». En effet, lors d'une transition, si un élément associé à l'ancienne tâche « disparaît » au profit d'un de ses voisins, associé à la nouvelle tâche, nous pouvons nous attendre à ce que cette transition induise une coupure franche sur cette liaison. Le cas échéant, le principe de transition différenciée peut être utilisé afin de fluidifier le passage d'une information à l'autre, ce type de transition ayant l'avantage de porter un sens plus fort qu'une simple transition paramétrée. Nous utilisons ce principe dans la section suivante, afin de proposer une transition visuelle entre deux modes d'un jeu vidéo (§ 5.5.4.1).

5.4.3 Intégration à une approche générative des interfaces

Dans une démarche méthodologique comparable, Moussa *et al.* (2000) s'appuient sur la définition de modèles pour la conception d'interface dans le contrôle de procédé industriel. Les auteurs proposent un outil, appelé ERGO-CONCEPTOR, permettant la génération semi-automatisée des interfaces. Cet outil intègre un ensemble de règles, s'appuyant sur des critères ergonomiques afin de diriger la génération des interfaces.

L'utilisation de ERGO-CONCEPTOR fait l'hypothèse que le concepteur peut « directement exploiter les données issues de l'analyse et la modélisation du système homme-machine considéré » (Moussa *et al.*, 2000). L'outil permet de spécifier le lien entre les différentes informations requises par l'opérateur puis de définir automatiquement, à partir de ces liens, des spécifications pour la génération de l'interface. Le processus de génération de l'interface s'appuie ensuite sur des critères ergonomiques relatifs à la présentation de l'information sur l'écran pour spécifier les différentes vues.

Nous pensons que l'utilisation de l'outil ERGO-CONCEPTOR est compatible avec la méthodologie SAMANTA. En effet, les données de modélisation issues de la première phase de notre méthodologie pourraient venir nourrir l'outil proposé par Moussa *et al.* (2000), et les résultats des analyses des transitions peuvent notamment venir ajouter des contraintes sur les liens entre ces données. De plus, lors de la spécification de l'interface, nous pouvons imaginer d'intégrer aux différentes règles ergonomiques les contraintes sur chaque nœud, exprimées sous la forme de méta-données, proposées dans le paragraphe précédent.

5.5 ILLUSTRATION DE LA MÉTHODOLOGIE : APPLICATION À UN JEU VIDÉO

En conclusion de ce chapitre, nous mettons en œuvre la méthodologie SAMANTA en l'appliquant au cas de la conception d'un jeu vidéo, Battlefield 3™. Dans un premier temps, nous présentons succinctement ce jeu vidéo puis, dans un second temps, nous en étudions une transition spécifique. Dans cette section, le terme « joueur » sera utilisé en lieu et place du terme « opérateur », plus propice aux discussions sur les systèmes complexes.

Notons que le contenu de cette section a fait l'objet d'une publication (Villaren *et al.*, 2012a).

5.5.1 Introduction

5.5.1.1 Présentation du jeu

Battlefield 3™ est un jeu vidéo de tir en situation de combat réaliste, édité en octobre 2011 par *Electronic Arts* (EA) et développé par *EA Digital Illusions Creative Entertainment* (DICE). Il fonctionne sur les ordinateurs munis de Microsoft Windows ainsi que sur les consoles Sony Playstation 3 et Microsoft XBOX 360.

Ce jeu vidéo en trois dimensions place le joueur au centre de l'action à travers une vue subjective à la première personne où il incarne un soldat au cœur d'un conflit armé contemporain dans différents théâtres d'opération. Deux types de jeu sont proposés à l'utilisateur :

- une campagne en solitaire, au cours de laquelle le joueur est immergé dans un scénario de jeu linéaire et se bat face à des joueurs contrôlés par l'ordinateur ;
- un jeu multijoueur, où le joueur est intégré à une équipe et coopère avec les autres joueurs connectés au serveur, contrôlés par d'autres êtres humains, afin de remplir des objectifs déterminés par le mode de jeu sélectionné, parmi les cinq proposés.

Conquête : match entre deux équipes de 32 joueurs maximum pour la capture et la protection de points de contrôle.

Ruée : match entre deux équipes de 32 joueurs maximum, où chaque équipe doit attaquer (ou défendre) un ensemble de relais de communication.

Ruée en Escouade : même mode de jeu que *Ruée* mais entre deux équipes de 4 joueurs maximum.

Match à mort en escouade : 4 équipes de 4 joueurs maximum s'affrontent et doivent éliminer chacune un nombre prédéfini de membres des autres équipes.

Match à mort par équipe : 2 équipes de 12 joueurs maximum s'affrontent et doivent éliminer chacune un nombre prédéfini de membres de l'autre équipe.

Au sein de chaque mode, le joueur doit choisir un poste parmi quatre classes possibles : *Assaut*, *Soutien*, *Ingénieur*, ou *Éclaireur*, auxquelles sont attribuées des armes et capacités différentes. Au cours de la mission, tous les joueurs ont la possibilité de piloter différents véhicules : transports rapides (jeeps et buggies), transports blindés, transports maritimes, hélicoptères, avions de chasse ou chars d'assaut. Leur position sur le champ de bataille est prédéfinie et ils y réapparaissent après avoir été détruits en jeu. Chaque mission se déroule sur une carte de jeu représentant un environnement particulier (ville ou désert au Moyen Orient, centre ville de Paris, île au milieu de l'océan Pacifique...).

Dans le cas de cette illustration de l'analyse des transitions, nous considérons un joueur d'une partie multijoueur en mode Conquête. Au démarrage de mission, ce joueur est assigné à une escouade de quatre membres appartenant à l'une des deux équipes en lice. Son objectif est, en collaboration avec les membres de son escouade et plus globalement de son équipe, de s'emparer et de protéger des points de contrôle répartis sur la carte de la mission.

5.5.1.2 Mise en œuvre de l'approche

Nous présentons dans la suite de cette section les modèles de tâche et de *Situation Awareness* construits afin de décrire l'activité d'un joueur de Battlefield 3™. Afin de construire ces modèles, nous avons collecté des données auprès de deux joueurs

réguliers de jeux vidéo dont Battlefield 3™ (et ses prédécesseurs dans la série des Battlefield). Nous nous appuyons également sur les données disponibles sur le site internet officiel du jeu, à l'adresse : <http://www.battlefield.com/battlefield3/>.

Le processus de modélisation et de validation s'est ainsi fait de manière continue, par des discussions et itérations successives. Les modèles obtenus sont présentés dans les sections 5.5.2 et 5.5.3. La transition considérée pour illustrer l'étape d'analyse est le passage d'un mode déplacement « à pied » (mode de déplacement par défaut) vers un mode de déplacement véhiculé, dans un char d'assaut, avec pour objectif de rejoindre et capturer les points de contrôle non occupés par l'équipe.

5.5.2 Modèle de tâches

Les tâches réalisées par le joueur sont décrites sur l'arbre de la figure 5.20. Ce cas d'application étant illustratif, nous nous sommes limités à définir les associations pour les tâches concernées par la transition analysée. Ces tâches sont visualisées par un petit cercle rouge sur la figure 5.20.

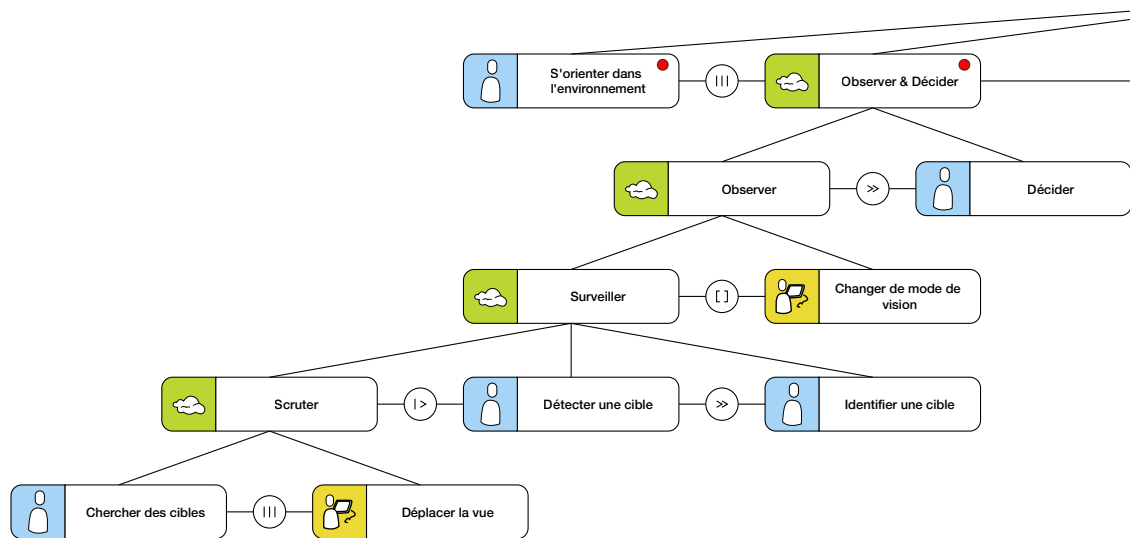
Ainsi, les quatre tâches en jeu dans cette analyse sont :

- Deux tâches descendantes de la tâche « Se déplacer », rassemblant l'ensemble des tâches décrivant les modes de déplacement disponibles dans le jeu :
 - La tâche « Se déplacer à pied » correspond au mode de déplacement par défaut (lorsque le joueur apparaît sur le champ de bataille, son avatar est un simple fantassin) ;
 - La tâche « Conduire un char » constitue la nouvelle tâche apportée dans la transition étudiée. Elle est exclusive par rapport à la tâche précédente : le joueur ne peut se déplacer à la fois à pied et conduire un char.
- En parallèle de ces deux tâches, deux tâches sont actives, quelque soit le mode de déplacement :
 - la tâche « S'orienter dans l'environnement » est présente en continu et correspond à l'action constante du joueur de se repérer par rapport à la carte, et à ses coéquipiers.
 - la tâche « Observer & Décider » correspond à la tâche de recherche d'objectif (les points de contrôle dans le scénario considéré).

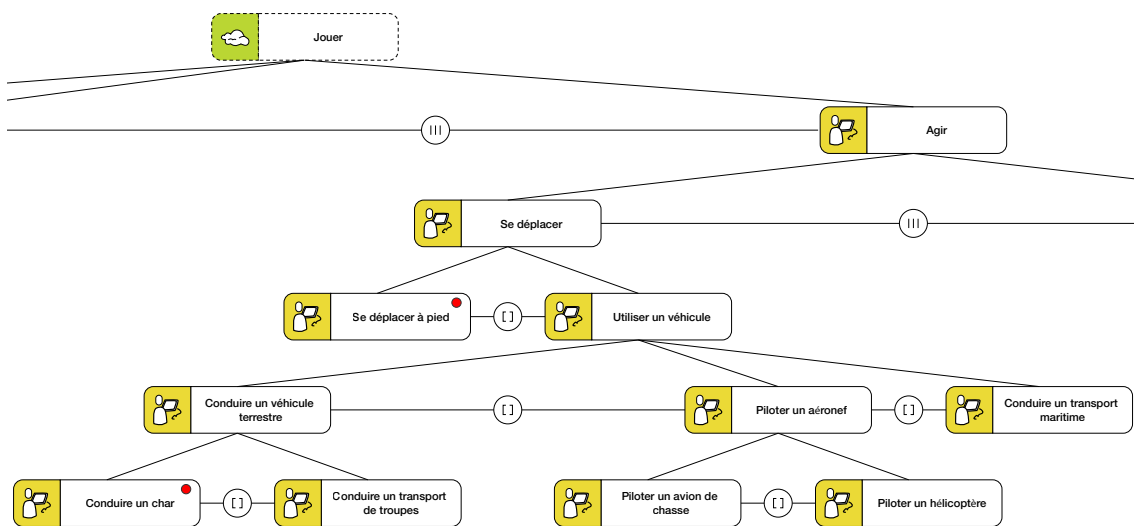
5.5.3 Définition des contextes associés aux tâches

Le modèle de *Situation Awareness* construit, présenté sur la figure 5.21, est composé de **54 nœuds** et de **80 arêtes**. Nous avons fusionné dans ce même graphe les contextes associés aux quatre tâches identifiées, chaque contexte est ainsi défini par une couleur différente :

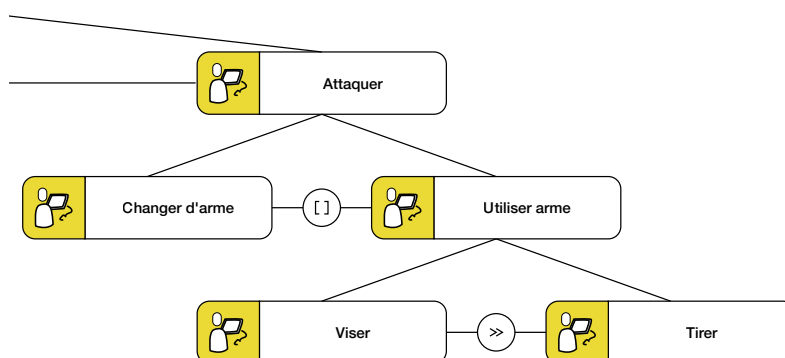
- Les éléments de situation associés à la tâche « Se déplacer à pied » sont visualisés en **saumon** ;



(a) partie gauche de l'arbre.



(b) partie centrale de l'arbre.



(c) partie droite de l'arbre.

FIGURE 5.20 : arbre CTT décrivant les tâches d'un joueur de Battlefield 3.

-
- The diagram is a semantic network for a tank simulation. The central node is 'joueur' (player), which is connected to numerous other nodes representing various game elements. The nodes are color-coded: green for vehicles and weapons, orange for player-related attributes, purple for team and target-related concepts, and blue for positional data. Relationships are indicated by labeled arrows such as 'est' (is), 'contient' (contains), 'appartient à' (belongs to), and 'possède' (owns). The network includes categories like 'type de véhicule' (vehicle type), 'véhicule' (vehicle), 'arme' (weapon), 'cible' (target), and 'équipe' (team).

168

Nous remarquons que près de la moitié des éléments du graphe global ne sont associés avec aucune des tâches considérées.

Ils correspondent principalement aux éléments relatifs aux ennemis, non pris en compte dans ces tâches, l'objectif de la mission étant la capture de points de contrôle et non l'identification d'ennemis. Toutefois, les éléments « ennemi » et « position de la cible » sont associés aux contextes des tâches « S'orienter dans l'environnement » et « Observer & Décider ». En effet, la position des ennemis, qui correspond à un accès visuel direct ou à la notification d'un coéquipier, influence sur l'orientation du joueur et sa prise de décision pour ses déplacements.

Les éléments relatifs à l'armement ne sont également pas inclus dans les contextes considérés, mise à part l'arme primaire, définissant l'arme courante (équipée par le joueur) et qui influence notamment sa vision et potentiellement sa vitesse (les armes lourdes ralentissent le joueur).

La figure 5.22 présente la façon dont sont visualisés quelques uns des éléments du graphe sur l'interface du joueur, en fonction de son mode de déplacement.

5.5.4 Analyse des transitions

Dans le scénario considéré, la transition étudiée consiste à comparer les contextes impliqués lorsque le joueur se déplace à pied et lorsqu'il conduit un char d'assaut. Cette transition est schématisée par la figure 5.23, où sont représentées les quatre tâches en jeu. Ce schéma illustre le produit de l'algorithme de recherche de la tâche parente commune aux deux tâches considérées dans la transition (section 5.3.1.1) – ici, la tâche « Se déplacer » – et permet d'identifier les tâches s'exécutant en parallèle des deux tâches composant la transition initiale (section 5.3.6).

Dans la suite de cette sous-section, nous présentons les résultats des analyses de la transition, en précisant lorsque les tâches parallèles ont été incluses ou non dans l'analyse.

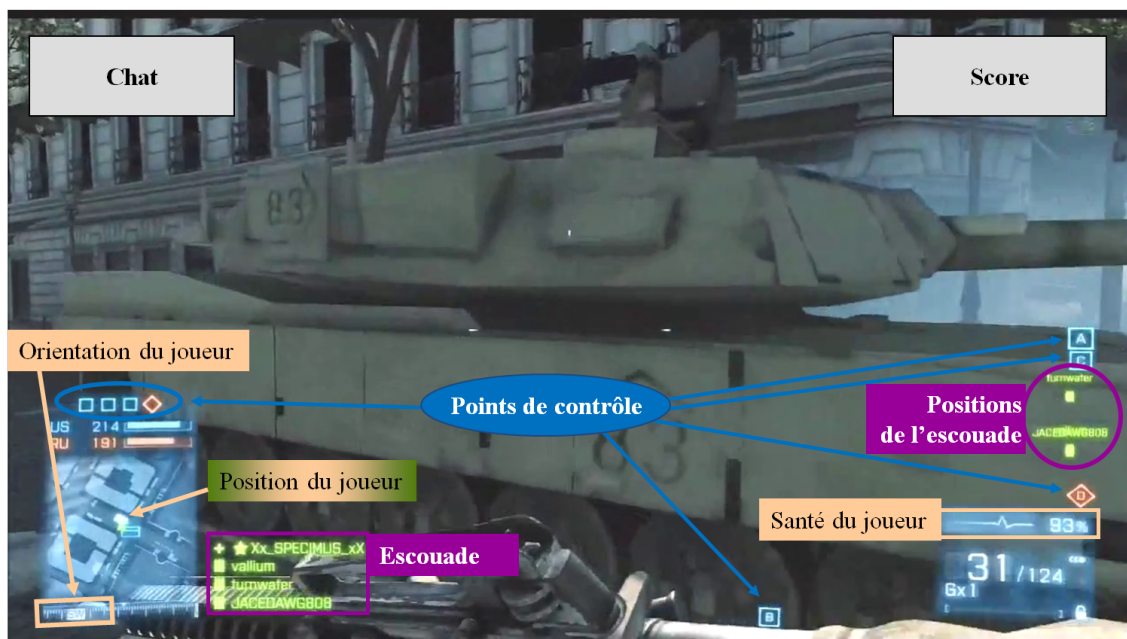
5.5.4.1 Analyse des statistiques

Nombres d'éléments associés à chaque contexte

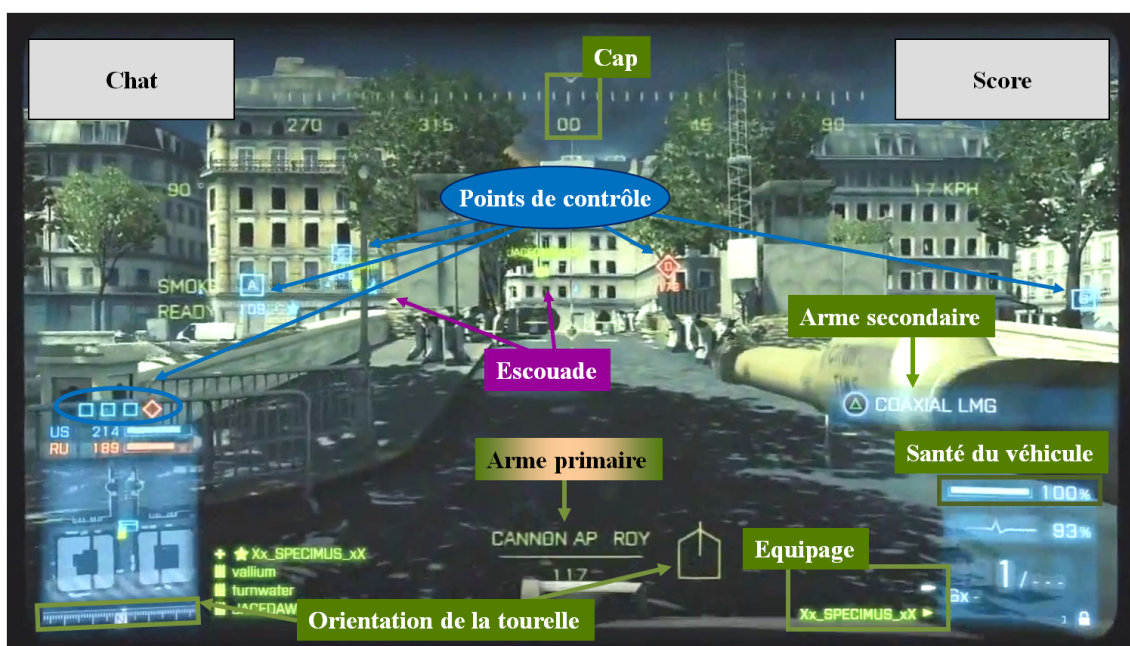
Parmi les 54 nœuds du graphe de situation, il y a 4 éléments propres à la tâche « Se déplacer à pied » : *joueur*, *santé*, *endurance*, *orientation du joueur*. Ces éléments sont importants lorsque le joueur se déplace à pied mais deviennent moins importants dans le cadre de la conduite de char : les déplacements sont centrés sur le char au lieu du fantassin, ce sont donc les paramètres liés à ce char qui deviennent importants.

Ainsi, il y a 11 éléments propres à la tâche « Conduire un char » : *véhicule*, *type de véhicule*, *char*, *cap*, *orientation de la tourelle*, *inclinaison de la tourelle*, *position du véhicule*, *santé du véhicule*, *équipement du char*, *chauffeur*, *mitrailleur*. Ces éléments correspondent aux nouveaux paramètres que le joueur doit prendre en compte lors de la transition : il s'agit de données concernant le type de char, d'armement, son

5.5. ILLUSTRATION DE LA MÉTHODOLOGIE : APPLICATION À UN JEU VIDÉO



(a) vue hors du chars : mode « marche à pied ».



(b) vue depuis le char, en tant que chauffeur/canonnier.

FIGURE 5.22 : captures d'écran d'une partie de Battlefield 3 multijoueur en mode Conquête.

état général ainsi que l'éventuel équipage qui le constitue et la place du joueur au sein de cet équipage. En effet, le joueur peut être le chauffeur, également en charge du canon, ou uniquement en charge de la mitrailleuse annexe. Un ordre de priorité

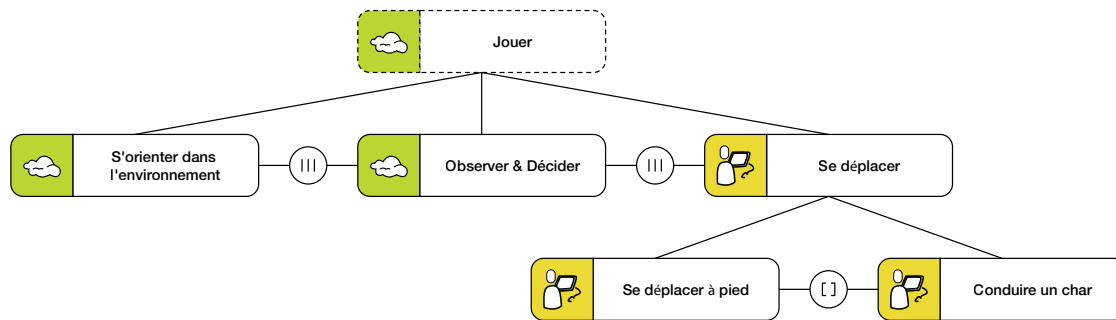


FIGURE 5.23 : schématisation de la transition analysée.

est imposé sur ce rôle (chauffeur puis mitrailleur) mais le joueur ne sait pas nécessairement à l'avance quel rôle il va obtenir.

Seuls deux éléments du graphe de SA sont partagés par les deux tâches. Il s'agit des éléments « arme primaire » et « position du joueur ». L'arme primaire est également un élément saillant au sens de Stanton, nous le traitons dans le paragraphe 5.5.4.3. En ce qui concerne la position du joueur, celle-ci ne change *a priori* pas lors de la transition, il n'est donc pas forcément nécessaire de mettre l'accent sur cet élément.

Lorsque l'on prend en compte les tâches parallèles dans l'analyse, cela apporte 14 éléments supplémentaires : *escouade, équipe, point de repère, bâtiment, route, étendue d'eau, membre, position du membre, ennemi, position de la cible, position du point, distance au point, point de contrôle, "propriétaire" du point*. La majorité de ces éléments appartient à l'un des trois sous-graphes communs, décrits dans le paragraphe 5.5.4.2.

Similarité - Score de Tversky

Nous considérons ces deux tâches comme symétriques, et fixons donc $\alpha = \beta = 1$ pour le calcul du ratio de Tversky. Pour cette transition, ce ratio est de 12 % de similarité entre les deux tâches. Ce faible score est expliqué par le fait qu'il n'y ait que 2 éléments partagés pour un nombre total de 17 éléments impliqués dans la transition.

Lorsque les tâches parallèles sont ajoutées au calcul de ce score, sa valeur est alors de 52 %. La différence par rapport au calcul du ratio sans ces tâches parallèles traduit un fort impact des tâches parallèles sur les éléments partagés par les deux tâches de la transition.

Taxinomie des transitions

Au sein de la taxinomie des transitions, le changement de tâches du déplacement à pied vers la conduite de véhicule est une transition simple, avec intersection des contextes (figure 5.24). Ceci est vrai que les tâches parallèles soient incluses à

l'analyse ou non. Comme nous l'avons décrit dans les paragraphes précédents, l'intersection dans le cas « sans tâches parallèles » est très petite : seuls deux éléments sont partagés.

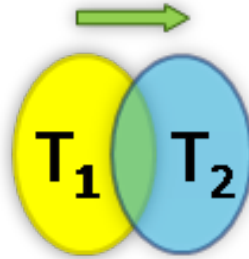


FIGURE 5.24 : la transition considérée vis-à-vis de la taxinomie. Il s'agit d'une transition simple, avec intersection de contextes.

Éléments « frontières »

La détection des éléments à la frontière entre les deux contextes n'est pas encore intégrée à l'outil. Néanmoins, à partir du graphe global coloré présenté sur la figure 5.21, nous détectons quatre nouveaux éléments, appartenant au contexte de la tâche « Conduire un char » (colorés en vert) et liés directement à des éléments associés à la tâche « Se déplacer à pied » (colorés en saumon). Il s'agit des éléments : *véhicule*, *équipage du char*, *position du véhicule* et *orientation de la tourelle*.

Les éléments *véhicule* et *position du véhicule* correspondent aux informations que le joueur connaît déjà puisque c'est lui qui choisit de monter dans ce véhicule. Nous pouvons considérer qu'il est « conscient » de leur existence.

Nous avons discuté de l'élément *équipage du char* dans la liste des nouveaux éléments ci-dessus. La mention du rôle du joueur doit être mise en exergue lors de la transition car il s'agit d'un élément non connu à l'avance *a priori* (sauf si l'information d'occupation du véhicule a été transmise auparavant par un coéquipier par exemple). Cette information est liée à l'information d'arme primaire (équipée par le joueur). Sur l'interface, le rôle du joueur est affiché en bas à droite à travers l'usage de pictogrammes accolés aux noms des deux joueurs de l'équipage.

Enfin, lors de la transition, un élément frontière intéressant est l'élément orientation de la tourelle. Il peut s'agir, selon le rôle du joueur, du canon du char ou bien de la mitrailleuse secondaire, dont la position diffère selon le modèle de char (mais dont le contrôle est indépendant de celui du canon principal). Ainsi, cette tourelle peut avoir une orientation différente de la caisse (le corps du char). Lors de la transition, le passage de l'orientation du joueur (vue à la première personne, regardant

« droit devant lui ») à celle de la tourelle peut impliquer un « saut » entre les deux, si le joueur n'entre pas dans le char dans l'axe de la tourelle. Sur l'interface actuelle du jeu, l'angle de la tourelle est visualisée à travers un pictogramme au centre en bas de l'écran ainsi qu'un compas en bas à droite. Nous pouvons imaginer d'ajouter une animation par exemple, consistant en un passage à la troisième personne (vue aérienne) afin de permettre au joueur de s'orienter dans son environnement et de capter cette nouvelle orientation. Ce type de transition a été par exemple étudié dans le cas des systèmes de drones par Draper *et al.* (2008).

Sur le principe des transitions différenciées (Schlienger *et al.*, 2006), nous pouvons ajouter un effet de « ralenti » sur la vitesse de rotation de la caméra lorsqu'elle passe de la vue du joueur à celle du char, afin d'indiquer la diminution de la capacité de rotation de la vue du joueur lorsqu'il passe en mode de conduite de char (la vue caméra tourne alors moins vite, elle est moins réactive).

5.5.4.2 Plus grands sous-graphes communs

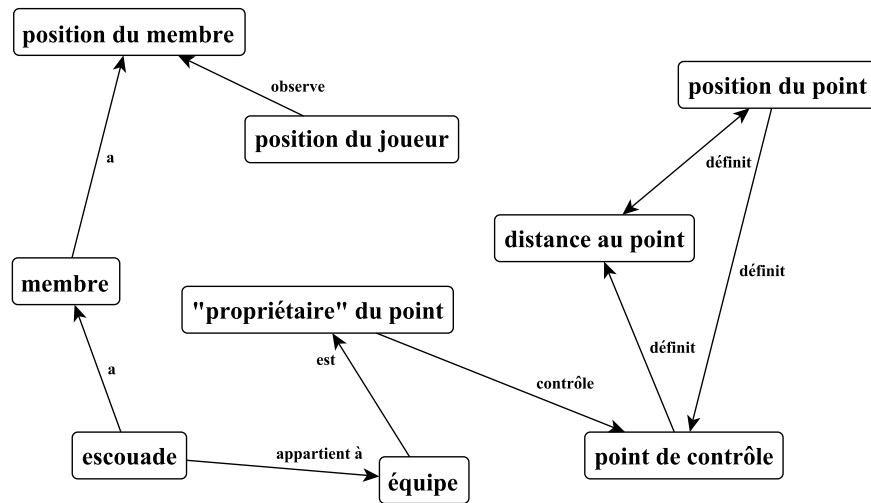
Comme nous l'avons vu dans le paragraphe précédent, les deux tâches composant cette transition ne partagent que deux éléments de situation. Ces deux éléments n'étant pas liés dans le graphe global modélisant la SA, l'extraction des plus grands sous-graphes communs à ces tâches ne fait ressortir que deux tâches uniques.

Néanmoins, lorsque l'on inclue les deux tâches parallèles à cette analyse, plusieurs sous-graphes appartenant à ces deux tâches combinées sont mis en avant (figure 5.25). Ces informations sont importantes car elles permettent au joueur de s'orienter dans l'environnement et fournir une continuité dans l'accès à ces données lors de la transition, ce qui facilite la reprise des objectifs en cours (la capture de points de contrôle dans ce scénario).

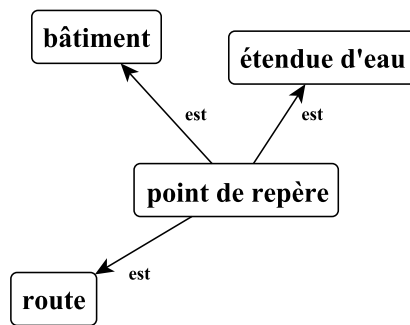
Le premier sous-graphe (figure 5.25a) regroupe les informations portant sur l'équipe et l'escouade auxquelles appartient le joueur ainsi que sur les points de contrôle de la carte et leur « propriétaire » (quelle équipe contrôle le point). La position de ces différents éléments (autres membres de l'équipe, plus particulièrement de l'escouade, et celle des points de contrôle) est visualisée sur l'interface à travers des pictogrammes de couleur situés sur les bords de l'écran ou au centre lorsque le joueur regarde dans leur direction. L'entrée dans le tank, en changeant l'orientation de la vue du joueur, va également changer la position de ces pictogrammes sur l'écran. Sachant que ces informations ne vont pas changer au cours de la transition, accompagner la transition peut permettre au joueur de se ré-orienter dans l'environnement plus rapidement et ainsi reprendre sa tâche de poursuite des objectifs en conséquence. Ceci peut par exemple être fait en indiquant au joueur où sont situés les points de contrôles non capturés et où des membres de l'équipe/escouade se dirigent.

Le deuxième sous-graphe (figure 5.25b) regroupe les informations sur l'environne-

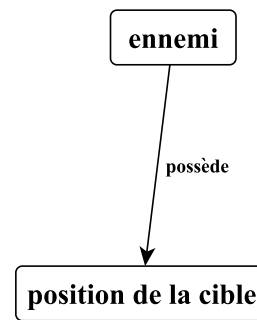
5.5. ILLUSTRATION DE LA MÉTHODOLOGIE : APPLICATION À UN JEU VIDÉO



(a) sous-graphe portant des informations sur l'équipe et les points contrôlés.



(b) sous-graphe portant sur l'environnement et les points de repères utilisés pour s'orienter.



(c) sous-graphe portant sur l'ennemi et sa position.

FIGURE 5.25 : les 3 plus grands sous-graphes communs de la transition, dont les éléments sont issus des tâches exécutées en parallèles.

ment du joueur, et particulièrement les caractéristiques utilisées par ce dernier pour se repérer (bâtiments, cours d'eau, routes...). Au niveau de l'interface, ces points de repères sont représentés sur la carte miniature affichée en bas à gauche de l'écran, centrée sur la position du joueur. Lors de la transition, l'orientation de cette carte peut changer en fonction de celle du char par rapport au joueur entrant. La présence de cette carte miniature permet au joueur d'acquérir un certain niveau de SA sur son environnement direct et de faire correspondre ce qu'il voit en trois dimensions avec cette autre vue en deux dimensions.

Enfin, le dernier sous-graphe mis en exergue lors de l'extraction (figure 5.25c) concerne les ennemis et leur position. Ces informations, que l'on peut également

considérer comme fixes lors de la transition (à l'instant où le joueur entre dans le char), sont également visualisées par des pictogrammes à l'écran, mais uniquement pour les ennemis en ligne de vue directe d'un coéquipier. Afin d'accompagner le joueur dans la transition, nous pouvons imaginer d'ajouter les informations sur la position des ennemis sur la carte miniature, ou en surbrillance sur l'écran afin d'attirer l'attention du joueur sur les dangers alentour.

5.5.4.3 Éléments saillants

Lors de la définition des contextes, nous n'avons pas défini d'éléments saillants pour l'une ou l'autre des tâches. Néanmoins, en reprenant la définition de la saillance selon Stanton, qui concerne les éléments de situation ayant 5 liens (ou plus) vers des éléments voisins, nous trouvons quelques éléments saillants pour les tâches impliquées dans la transition.

Un élément de situation partagé par les deux tâches est saillant au sens de Stanton. Il s'agit de l'élément *arme primaire*. Cet élément est important pour le joueur car il définit en grande partie sa stratégie lors de ses déplacements. En effet, lorsqu'il se déplace à pied, son arme aura un impact sur sa vitesse de déplacement et sur sa stratégie d'approche des objectifs par exemple. Dans le cas de la conduite de char, le type d'arme (dépendant du modèle) va influencer la stratégie de jeu : le temps de rechargement et la mobilité du canon sont différents selon les modèles par exemple. Lors de la transition, un message temporaire et visible indiquant quelle est l'arme actuellement équipée pourrait permettre d'améliorer la SA du joueur à ce sujet (en plus du nom de l'arme, affichée sur l'interface en bas, figure 5.22b).

Pour la tâche « Se déplacer à pied », un autre élément est saillant. Il s'agit de l'élément *joueur*, correspondant au fait que dans ce cas de figure, la vue est centrée sur le joueur.

Pour la tâche « Conduire un char », quatre autres éléments sont saillants : *véhicule*, *type de véhicule*, *char*, *équipage du char*. Il s'agit de quatre éléments en rapport avec la conduite. Les trois premiers, bien que saillants, doivent être bien intégrés par le joueur lors de la transition puisqu'ils se rapportent au type de véhicule considéré, choix que le joueur a effectué *a priori*. En ce qui concerne l'équipage du char, nous avons traité le cas de cet élément lors de la discussion sur les éléments frontières. Le fait qu'il soit saillant et qu'il s'agisse d'un nouvel élément renforce l'importance de cet élément et la pertinence d'accompagner son « apparition » dans le champ du joueur lors de cette transition.

5.5.5 Synthèse du cas applicatif

Nous avons présenté dans cette section un exemple illustratif d'application de la méthodologie SAMANTA à l'analyse d'une transition dans un jeu vidéo grand public. Il s'agit, comme nous l'avons indiqué en introduction, d'un exemple purement

illustratif permettant de mettre en avant sur une interface simple les quelques moyens qui pourraient être mis en œuvre afin de soutenir une transition dans ce jeu vidéo. Dans le cas présenté, l'interface finale existait déjà et nous n'avons fait qu'ajouter quelques propositions afin d'améliorer la prise en compte d'une transition spécifique.

Cependant, cet exemple nous a permis de présenter le principe de la modélisation et de l'analyse des transitions pour un cas d'étude dont l'appréhension est relativement aisée. Nous proposons dans le chapitre 7 d'appliquer la méthodologie à une situation d'étude plus complexe, la conception d'une station de contrôle de drone.

5.6 SYNTHÈSE DU CHAPITRE 5

Dans ce chapitre, nous avons présenté les différentes étapes de la méthodologie SAMANTA divisée en deux phases – modélisation et analyse des transitions – et son application au cas d'un jeu vidéo de tir.

L'analyste joue un rôle très important à toutes les étapes de cette méthodologie car il doit s'assurer de la logique globale des modèles, et des résultats des analyses. A l'issue de l'application de cette méthodologie par l'analyste, expert en facteurs humains et ergonomie, les recommandations émises afin de prendre en compte l'impact des transitions doivent permettre de guider la conception de ces interfaces.

En particulier, dans le cadre d'un processus centré sur les utilisateurs, il fera appel aux opérateurs afin de valider les modèles mais aussi, dans le cadre d'un processus de conception plus global dans lequel s'intégrerait la méthodologie SAMANTA, afin de valider les concepts proposés pour soutenir les transitions. Cette validation finale peut être effectuée sur la base de maquettes, de simulation ou de prototypes sur lesquels l'opérateur et l'analyste vont simuler, lors de tests utilisateurs, l'opération du système afin de valider ou non les décisions prises.

Ainsi en proposant un fonctionnement « classique » basé sur le trio analyse / conception / validation, la méthodologie SAMANTA peut s'intégrer à d'autres méthodes de conception de systèmes agissant à un niveau plus large que la seule étude des transitions, et bénéficier des ressources issues de ces travaux, tout en leur faisant bénéficier des résultats de l'analyse des transitions.

Néanmoins, SAMANTA présente la limite d'être conçue pour des analystes ayant une expertise en ergonomie et en Facteurs Humains. Il est important lors de sa mise en œuvre de s'assurer de la compréhension par l'analyse des concepts sous-jacents, notamment en termes de *Situation Awareness* et de modélisation de l'activité des opérateurs, afin d'éviter toute dérive.

Afin de soutenir l'application de cette méthodologie par un ergonome / analyste, et permettre son opérationnalisation sur de futurs projets concrets, nous avons développé un outil soutenant l'analyse des transitions en s'appuyant sur les modèles construits par l'ergonome.



TROISIÈME PARTIE : MISE EN ŒUVRE DE L'APPROCHE, CONTRIBUTION PRATIQUE

6

Outillage de la méthodologie SAMANTA

Dans ce chapitre, nous introduisons l'outil conçu dans le cadre de cette thèse afin d'appuyer l'usage de la méthodologie SAMANTA (*Situation Awareness ANalysis for Transition Amelioration*) par l'analyste. Cet outil support se présente sous la forme d'une application web dynamique, fonctionnant à la fois sur les supports mobiles (tablettes tactiles) et les ordinateurs personnels.

6.1 DÉMARCHÉ

La méthodologie SAMANTA a été conçue avec pour objectif sa mise en œuvre par un(e) ou plusieurs analystes ayant une expertise dans le domaine de l'ergonomie et des facteurs humains, en charge de la conception de systèmes complexes.

Comme nous l'avons vu dans le chapitre précédent, l'approche adoptée, centrée sur les utilisateurs, nécessite d'aller à la rencontre des opérateurs, ou d'experts du domaine, afin de concevoir un système correspondant au mieux à la réalité opérationnelle. Ceci nécessite, de la part de l'analyste, une grande mobilité et des allers/retours fréquents avec les différents acteurs de la conception.

De plus, la complexité des systèmes conçus induit généralement la prise en compte d'un nombre important de tâches et de leurs contextes, ce qui se traduit lors de l'opérationnalisation de la méthodologie, par un travail fastidieux de modélisation (étapes 2 et 3). De surcroît, l'analyse de transitions, réalisée « à la main » par l'analyste, est d'autant plus prône à l'erreur qu'elle nécessite la prise en compte d'un grand nombre d'éléments.

Devant ces constats, il nous semble intéressant de proposer la mise en place d'un outil informatique pour la modélisation et l'analyse des transitions afin de « décharger » l'analyste des tâches redondantes, fastidieuses et automatisables liées à ces étapes, tout en servant de support aux échanges avec les utilisateurs lors des phases de validation. Ces analystes, qui devront avoir un minimum de connaissances des concepts et principes en jeux, pourront ainsi s'appuyer sur cette outil pour approfondir leur étude et ainsi s'abstraire du couple papier/crayon. Caffiau *et al.* (2008) mettent en avant le fait qu'un tel outil peut venir en support de l'apprentissage de la méthodologie qu'il soutient. En effet, ils font le constat que la grande majorité des étudiants d'un cours sur K-MAD préfère utiliser le logiciel K-MADe pour réaliser

les exercices de modélisation du cours plutôt qu'une feuille et un crayon.

Cette réflexion sur l'intérêt d'un outil mis à disposition des ergonomes et experts en facteurs humains a su trouver un écho favorable auprès des experts côtoyés tout au long de cette thèse, dont la pratique quotidienne de l'ergonomie et des facteurs humains pour la conception de systèmes complexes est marquée par la recherche de la simplification des processus d'analyse et de leur structuration, les outils utilisés n'étant pas toujours adaptés.

6.2 CHAMP D'APPLICATION DE L'OUTIL DÉVELOPPÉ

Afin de soutenir l'usage de la méthodologie SAMANTA par les ergonomes, l'outil développé se doit d'être simple d'utilisation et doit être utilisable tant en situation de mobilité, afin d'aller au contact des utilisateurs et valider les modèles produits, que sur un ordinateur de bureau, pour le travail quotidien de modélisation et d'analyse.

Les modèles retenus dans la méthodologie SAMANTA s'appuyant sur des outils déjà connus des ergonomes, et utilisés selon leurs besoins, nous nous focalisons sur les « nouveautés » apportées par la méthodologie et nous appuyons sur l'usage amont de ces outils pour la construction des modèles (étape 2 de la méthodologie).

- La construction de l'arbre CTT est appuyée par l'éditeur CTTE, mis à disposition et maintenu par l'équipe à l'origine de la notation (voir aussi le paragraphe 6.3.2.1).
- La construction du réseau propositionnel décrivant les éléments de *Situation Awareness* peut se faire à travers un éditeur de graphe générique. Nous utilisons et recommandons l'utilisation de l'application *yEd Graph Editor* (§ 6.3.2.2), compatible avec de nombreux formats de description des graphes et jugé simple d'utilisation par les ergonomes de l'équipe.

Nous nous focalisons ainsi sur la prise en charge des étapes 3 et 4 de la méthodologie, à savoir la définition des contextes et l'analyse des transitions. En s'inscrivant dans une « chaîne outillée » soutenant les étapes 2 à 4 de la méthodologie, l'objectif est de permettre à l'analyste, en s'appuyant sur les connaissances extraites dans l'étape 1, de créer les deux modèles introduits dans la méthodologie (étape 2) à partir d'outils annexes. Dans un deuxième temps, l'analyse doit pouvoir définir, pour chaque tâche de l'arbre, le contexte associé (étape 3) et lancer une analyse des transitions (étape 4), en utilisant l'outil développé.

Une composante de mobilité est imposée afin de permettre la validation par les opérateurs des modèles et des contextes définis pendant l'étape 3, avant l'analyse des transitions (étape 4). La figure 6.1 résume les outils associés aux étapes de la méthodologie.

Afin de soutenir les étapes 3 et 4 de la méthodologie, tout en étant compatible avec les formats de données issus de l'étape 2 (section 6.3.2), nous avons spécifié un ensemble de fonctions, décrites en annexe B. Dans ce chapitre, nous présentons les

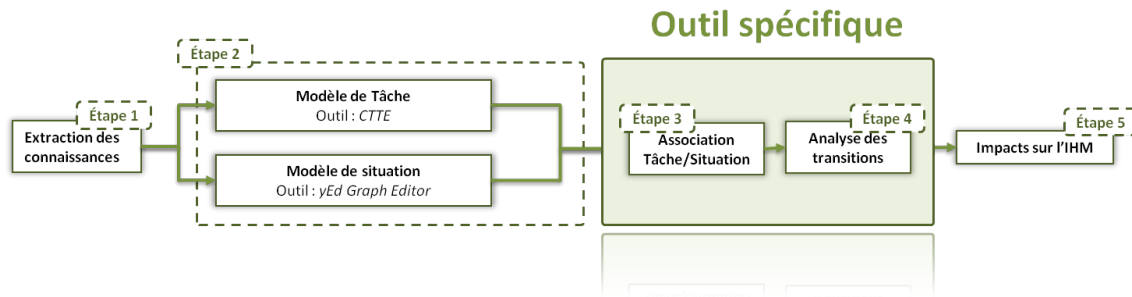


FIGURE 6.1 : les étapes 2 à 4 de la méthodologie et les outils associés.

choix techniques réalisés afin de répondre aux critères de mobilité, compatibilité et simplicité imposés par l'activité des ergonomes, utilisateurs finaux de l'outil, puis détaillons les modèles de données sous-jacents. Nous concluons ce chapitre sur une description des interfaces de l'outil et présentons quelques perspectives de développement.

6.3 SPÉCIFICATIONS TECHNIQUES DE L'OUTIL

6

Cette section introduit les choix technologiques et structurels effectués afin de répondre aux besoins définis dans la section précédente. Dans un premier temps, nous détaillons les critères ayant motivé notre choix de la technologie Web pour le développement de notre outil. Ensuite, nous décrivons les données d'entrées fournies à l'outil, suivies de la description de la structure des données manipulées au sein de l'application. Le contenu de cette section est complété par les descriptions techniques des formats et bibliothèques utilisées, présentées en annexe C.

6.3.1 Choix de la technologie

Afin de choisir le type de technologie logicielle mise en place pour le développement de notre outil, nous avons défini un ensemble de critères, définis par rapport aux utilisateurs finaux, les experts-ergonomes, pour qui l'application doit être simple d'installation, de mise à jour et d'utilisation.

6.3.1.1 Définition des critères

Ainsi, les critères de choix retenus sont les suivants :

Compatibilité multi-plateforme

Afin d'assurer la diffusion et l'accès à l'outil quelque soit le support à disposition de l'utilisateur, la technologie utilisée devra être compatible avec les différentes plateformes et systèmes d'exploitation utilisés par les utilisateurs-ergonomes, à savoir :

- Les ordinateurs (de bureau ou portable) faisant fonctionner les systèmes d'exploitation *Windows* (toutes versions depuis *Windows XP*) ou *Mac OS X*, qui constituent les deux systèmes d'exploitation les plus répandus¹ ;
- Les appareils mobiles de type tablettes tactiles, faisant fonctionner *Android OS* ou *iOS* (le système d'exploitation de l'iPad). L'usage de tels support permettra notamment d'utiliser l'outil en situation de mobilité, lors de déplacement auprès des opérateurs (en particulier pour la validation des modèles et associations, suite à l'étape 3).

Fonctionnement hors ligne

L'outil développé doit pouvoir fonctionner sans connexion à Internet, afin de répondre aux situations de mobilités auxquelles peuvent être confrontés les utilisateurs-ergonomes, par exemple, lors des entretiens avec les utilisateurs. Néanmoins, certaines fonctionnalités, comme l'installation ou la mise à jour, pourront être accessibles uniquement en mode « connecté ».

Pérennité du projet

Afin de faciliter d'éventuels développements futurs par d'autres personnes, et ainsi assurer une pérennité du projet, les langages et technologies sélectionnés devront être « classiques ». Nous éviterons ainsi d'utiliser des langages de programmation peu connus ou mal documentés.

Rapidité de développement

Afin d'assurer un temps de développement raisonnable, les langages et technologies de développement utilisés seront connus du doctorant.

Nous identifions deux familles de solutions envisageables afin de répondre à ces besoins :

- Le développement d'une application native, installée sur la plate-forme de l'utilisateur et pouvant se connecter à Internet, et donc à un serveur distant, pour effectuer les mises à jour.
- Le développement d'une application internet riche (*Rich Internet Application* ou RIA en anglais), accessible à travers un navigateur Web et pouvant être utilisée localement à la plateforme (après connexion au service).

Ce type d'application peut s'appuyer sur plusieurs types de technologie

- Les technologies dites « de *plug-in* », nécessitant l'installation d'une application tierce pour le lancement de l'outil ;
- Les technologies HTML5 et Javascript, permettant de construire une application web dynamique, déployée sur un serveur distant et accessible en ligne sans installation préalable requise à travers un navigateur récent. Les mises à jour sont alors transparentes et s'effectuent à chaque connexion au serveur.

Nous discutons dans les paragraphes suivants de ces solutions envisagées, en fonction des critères définis pour le développement de l'outil.

1. représentant plus de 99 % des systèmes d'après le site <http://www.netmarketshare.com/>

6.3.1.2 Application native

L'outil développé sous la forme d'une application native se présenterait sous la forme d'un logiciel à installer sur la plateforme de l'utilisateur, à partir d'un fichier téléchargé ou transféré à l'utilisateur, dont le format dépend de la plateforme.

Pour ce type d'usage, l'outil fonctionne par défaut en mode local et ne nécessite donc pas de connexion à internet pour son fonctionnement, satisfaisant le critère de **fonctionnement hors-ligne**. Si nécessaire, l'application peut se connecter à un serveur distant afin d'enregistrer des données en ligne mais par défaut, l'utilisateur-ergonome peut enregistrer les données directement sur le support utilisé.

Néanmoins, même si un même langage de programmation peut être utilisé afin de produire un fichier exécutable **compatible** pour les systèmes d'exploitations « bureautiques » (familles Windows et Mac Os X), certaines spécificités propres à ces systèmes peuvent nécessiter des adaptations du code, ou l'ajout de bibliothèques particulières, engendrant des développements supplémentaires, et impactant la **rapidité de développement**. De plus, le développement d'une application mobile pour tablette est basé sur des langages différents de ceux utilisés pour la bureautique et nécessitent donc de développer en parallèle différents outils. Le tableau 6.1 présente les différences en termes de développement pour les systèmes considérés. Ces différences sont un frein au critère de **rapidité** mais également à celui de **pérennité** du développement car cela demande de maintenir trois codes sources en parallèle, tâche fastidieuse et nécessitant des ressources de développement importantes.

TABLEAU 6.1 : langages de programmation des applications natives sur ordinateur et supports mobiles

Support	OS	Langages
Ordinateur	Windows / MacOS X	Java ou C++
Support mobile (tablettes)	iOS (iPad)	Objective-C
	Android OS	Java Android

Enfin, quelque soit le support de l'utilisateur (ordinateur ou tablette), il est nécessaire d'installer localement un logiciel sur le système, sous forme d'application mobile, de machine virtuelle Java, ou d'application à part entière. Cette procédure doit être répétée à chaque fois qu'une mise à jour est disponible, et ce sur tous les supports utilisés. De plus, elle impose à l'utilisateur faisant cette mise à jour d'avoir

les droits d'administration nécessaires sur le support, ce qui n'est pas toujours le cas dans un milieu professionnel. Ces considérations sur le processus de maintien à jour des applications mettent un frein au critère de **compatibilité**.

6.3.1.3 Application Internet Riche

Les applications internet riches (RIA) se présentent sous deux formes technologiques :

- Un module complémentaire (*plug-in*) est installé sur le système d'exploitation de l'utilisateur et se lance au sein de son navigateur Internet. Il s'agit par exemple de modules développés à partir des technologies *Java*, *Adobe Flash/Flex* ou *Microsoft Silverlight*. L'utilisateur doit alors se connecter à un serveur distant où est déployé le code de l'outil, afin de le rapatrier localement et de le lancer à travers le module installé.
- L'outil est développé sous la forme d'une application web dynamique à partir des technologies HTML 5 et Javascript. L'utilisateur doit alors se connecter à une page web particulière, à partir d'un navigateur compatible (la plupart des navigateurs récents) puis utiliser l'application au sein de son navigateur (avec ou sans connexion).

Dans les deux cas de figure présentés, l'application ainsi développée **fonctionne en mode « hors ligne »**. Dans le cas de l'installation d'un module, l'application peut être généralement téléchargée localement puis lancée depuis le bureau en utilisant ce même module (à la manière d'une application native). Dans le cas du fonctionnement en mode « web dynamique », les nouvelles fonctionnalités du standard HTML5 permettent au client de stocker certains fichiers en local pour une utilisation hors ligne.

Ce type de fonctionnement présente également l'avantage de ne nécessiter le développement et le maintien que d'un code source, commun à toutes les plateformes (avec d'éventuelles adaptations pour la prise en charge des interfaces tactiles), répondant ainsi aux critères de **pérennité** et **rapidité de développement** .

Néanmoins, dans le cas d'un fonctionnement en mode *plug-in*, les modules cités (Java, Flash et Silverlight) ne sont **pas compatibles** avec tous les supports. En particulier, les tablettes iPad, représentant plus de 60 % des tablettes du marché², sont incompatibles avec ce type de modules. De plus, ce type de module nécessite une installation et des mises à jour régulières du module (dépendant d'un éditeur tiers) qui nécessitent de posséder les droits d'administration adéquats, pouvant freiner la mise en place de nouvelles versions.

2. Source : <http://www.netmarketshare.com/>

Ces inconvénients sont absents du mode de fonctionnement « web dynamique », l'utilisateur consultant l'application au sein d'une page web directement depuis son navigateur, sans installation supplémentaire. Il nécessite néanmoins un *navigateur compatible* (annexe C.2).

6.3.1.4 Synthèse concernant le choix de la technologie

Ainsi, parmi les alternatives présentées, le format d'application web dynamique, répond à l'ensemble des critères définis : compatibilité multi-plateforme, facilité et rapidité de développement/maintenance et fonctionnement hors-ligne.

En effet, ce type de développement permet d'assurer la compatibilité avec les différents supports choisis (ordinateurs et tablettes). Les technologies en jeu, à savoir HTML 5, Javascript et la description de graphique SVG, sont compatibles avec les navigateurs récents de ces supports, réduisant le travail de développement à une version unique tout support. La seule différence en termes de développement réside dans la prise en charge des interactions tactiles sur tablette afin d'adapter les interactions à la souris de la version non-mobile et proposer des moyens naturels d'interaction tactile.

Enfin, bien que l'utilisation de l'application web dynamique se fasse à travers un navigateur et nécessite de se connecter à un serveur, les fichiers nécessaires à son fonctionnement peuvent être installés en local à travers différentes méthodes. Dans le cas de l'usage sur tablette tactile, cette mise en cache nécessite forcément une connexion à Internet (l'accès à l'espace de stockage des tablettes étant moins naturel que pour les ordinateurs de bureau) mais les nouveautés introduites par la version 5 du standard HTML permettent de préciser au navigateur client les informations qu'il doit stocker localement³.

6.3.2 Données d'entrée de l'outil

Comme précisé en introduction et dans la description fonctionnelle de l'outil (section B), les données d'entrées de l'outil développé sont les modèles de tâche et de SA, construits à partir des applications supports CTTE et *yEd Graph Editor*, afin de servir de base pour le travail de définition des contextes associés aux tâches (étape 3 de la méthodologie).

Les formats de fichiers retenus pour la description de ces modèles sont basés sur le langage XML (*Extensible Markup Language*), pris en charge nativement par le langage *Javascript* utilisé pour le développement de notre application et ne nécessitant ainsi pas de développements supplémentaire (fonction d'ouverture / lecture / fermeture de fichiers) pour importer les modèles construits.

3. voir <http://www.w3.org/TR/html5/offline.html>

6.3.2.1 Description des arbres de tâches CTT.

Les tâches de l'opérateur dans le cadre de la méthodologie SAMANTA sont formalisées à travers la notation *ConcurTaskTree* (CTT) proposée par Paternò *et al.* (1997). Cette notation est appuyée par l'application CTTE (*CTT Environment*), développée par l'équipe de Paternò (Mori *et al.*, 2002). Elle se présente sous la forme d'une application bureautique, développée en Java et peut être téléchargée à l'adresse <http://giove.isti.cnr.it/tools/CTTE/>. Compatible avec les systèmes d'exploitation supportant la technologie Java, cet outil permet d'exporter les arbres de tâche créés dans un format binaire fermé (extension *.ctte*) mais aussi sous la forme d'un fichier XML. La description de ce fichier est formalisée par un schéma DTD (*Document Type Definition*) spécifique, fourni dans la documentation et décrit en annexe 6.3.2.1. De plus, l'application CTTE est toujours maintenue par ses distributeurs et de nouvelles versions sont mises à jour régulièrement ⁴.

6.3.2.2 Graphe des éléments de situation

Le formalisme de représentation de la *Situation Awareness* qui a été retenu dans le cadre de la méthodologie SAMANTA est celui des réseaux propositionnels de la théorie *Distributed Situation Awareness* (DSA) de Stanton *et al.* (2006).

La représentation des réseaux propositionnels se fait par la construction d'un graphe orienté dont les nœuds et les arêtes peuvent être étiquetés. Plusieurs formats de description des graphes existent et sont utilisés par diverses communautés, scientifiques, industrielles ou hobbyistes.

Parmi ces formats, nous retenons le format *GraphML*, basé sur le langage XML, issu d'un travail de standardisation mené par la communauté du *Graph Drawing*. Il permet de décrire des graphes sous un format largement répandu (XML), fournissant une base commune standardisée pour la description des graphes. L'extensibilité de XML laisse à chaque éditeur de logiciel une certaine liberté pour ajouter à GraphML les données propres à l'application tout en respectant le formalisme défini, et décrit au sein d'un schéma DTD propre. Le site <http://graphml.graphdrawing.org/> présente l'ensemble des ressources nécessaires à l'usage de ce langage.

Afin d'appuyer la construction des graphes, nous avons utilisé le logiciel *yEd Graph Editor*, disponible gratuitement à l'adresse <http://www.yworks.com/>. Cet outil permet de formaliser le réseau propositionnel de SA, et propose notamment des options de visualisation et d'auto-agencement des nœuds permettant de rendre le graphe produit plus lisible avant de l'exporter sous différents formats. Le format GraphML pris en charge par cette application est décrit en annexe 6.3.2.2.

4. la version 2.6.0 a été publiée le 31 juillet 2012.

6.3.2.3 Synthèse

Ainsi, les données d'entrée de l'outil soutenant la méthodologie SAMANTA sont toutes deux au format XML (bien que l'extension du fichier issu de *yEd Graph Editor* soit différent) et pourront être traitée par l'outil (figure 6.2). Ces données d'entrée seront ensuite traitées par les scripts de l'application afin de nourrir les structures de données et de construire les modèles de la méthodologie.

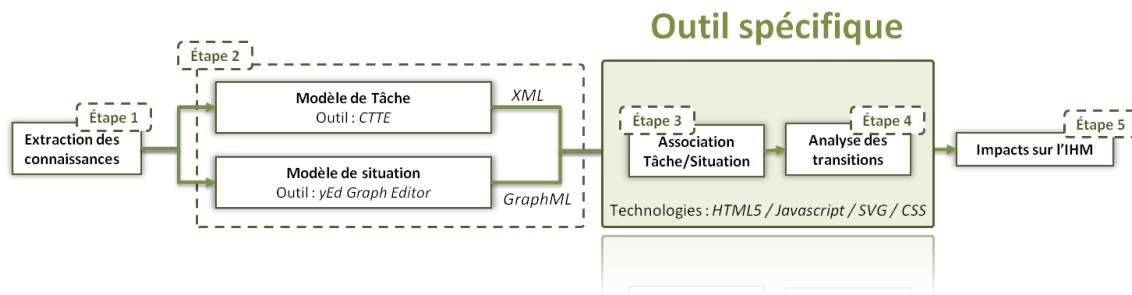


FIGURE 6.2 : les technologies soutenant les étapes 2, 3 et 4 de la méthodologie SAMANTA.

6.3.3 Architecture de l'application

L'application web dynamique développée et présentée dans ce chapitre s'appuie sur différents modèles de données, afin de représenter le modèle de tâches (arbre CTT), le modèle de SA (graphe orienté) et les contextes définis pour l'analyse des transitions. Nous présentons dans cette sous-section un aperçu de la structure adoptée afin de représenter ces données au sein de l'application.

6.3.3.1 Note sur le langage Javascript

Javascript est un langage de programmation par script, historiquement utilisé au sein des navigateurs Internet (côté client) afin de permettre la manipulation dynamique de leur contenu (à travers l'accès au modèle objet de document, le DOM) et enrichir les interactions avec ces pages. Des versions de Javascript existent également pour l'exécution côté serveur, permettant de remplacer les langages classiques utilisés pour servir des pages Web aux clients⁵.

Javascript autorise la programmation itérative, fonctionnelle ainsi que la programmation orientée objet. Javascript est un langage « tout objet » mais à l'inverse d'autres langages comme Java ou PHP, la définition des objets manipulés dans le cadre d'une application web dynamique ne repose pas sur la déclaration de classes et

5. citons notamment le projet NodeJS (<http://nodejs.org/>)

leur instantiation, mais sur la définition de fonctions « constructeurs » permettant d'instancier ces objets en mémoire.

Ainsi, dans la suite de cette section, notamment dans la description des modèles de données présentés dans le paragraphe suivant, nous utiliserons la notion d'*objet* manipulé plutôt que celle de *classe*, afin de s'accorder avec le langage de programmation choisi.

6.3.3.2 Les modèles de données

Dans ce paragraphe, nous définissons les modèles mis en place dans l'architecture de l'application. La construction des modèles est basée sur les objets manipulés dans le cadre de la méthodologie. Afin de décrire ces modèles, nous nous inspirons du langage UML (*Unified Modeling Language*), en particulier de la notation de diagramme de classe afin de décrire les attributs principaux de chaque objet modélisé et les relations entre ceux-ci. Notons que l'objectif de cette description est d'apporter une compréhension des liens entre ces différents modèles et non de décrire de manière exhaustive les attributs et méthodes attachés aux objets modélisés, ni de formaliser l'ensemble des classes utilisées au sein de l'application.

Modèles représentant l'arbre CTT

L'objet décrivant l'arbre au sein de l'application (objet *Tree*) est composé d'objets « Tâche » (objet *Task*) et « Transition » (objet *Transition*), comme représenté sur la figure 6.3. La relation de composition est visualisée par les diamants noirs aux extrémités des liens entre ces trois objets.

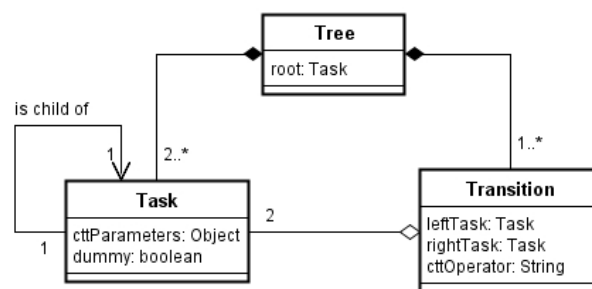


FIGURE 6.3 : diagramme UML de classe de représentation de l'arbre CTT.

Chaque arbre est ainsi composé d'un minimum de deux tâches, plus la tâche *root*, indiquée séparément car elle ne prend pas part à l'analyse des transitions et à la définition des contextes. Nous considérons qu'un arbre est composé d'au moins une transition.

Afin de définir une transition, il est nécessaire d'y associer deux tâches. Dans la description de l'objet *Transition*, ces deux tâches sont précisées de façon redondante avec la relation d'agrégation, ceci afin de mettre en exergue le caractère ordonné

de chaque transition à travers la dualité *leftTask/rightTask*. Une transition est également définie par un identificateur CTT (représenté par une chaîne de caractère *cttOperator*).

Notons qu'il peut y avoir dans l'application un nombre d'instances de *Transition* supérieur au nombre de transitions directement visualisé sur l'arbre CTT. En effet, ces transitions peuvent être définies par l'analyste selon les modalités décrites dans la section 5.3.1, par exemple pour deux tâches n'ayant pas le même parent. On utilisera alors les algorithmes définis dans la section 5.3.1 afin de définir l'identificateur CTT séparant les deux tâches impliquées, en conservant l'ordre de ces tâches (tâche de gauche et tâche de droite).

Chaque tâche de l'arbre possède une tâche parente (visualisée par la relation *is child of*), exceptée la tâche racine. Afin de préciser quelles sont les tâches « factices » (construites pour lever les ambiguïtés sur l'ordre des relations temporelles), un attribut booléen *dummy* a été ajouté. Les attributs CTT issus de la description des tâches dans le fichier XML importé sont regroupés sous la forme d'un objet JSON *cttParameters* (annexe C.4.2). Il décrit notamment des informations sur la catégorie de la tâche (abstraite, utilisateur, système, interaction), son nom, sa description, son caractère optionnel et/ou itératif, etc.

Modèles représentant le graphe de situation

Le modèle du graphe orienté (objet *Graph*) est une composition de « Nœuds » (objet *GraphNode*) et « d'arêtes » (objet *GraphEdge*). Un seul graphe existe par projet et est composé au minimum d'un nœud. La figure 6.4 représente les relations entre ces trois objets.

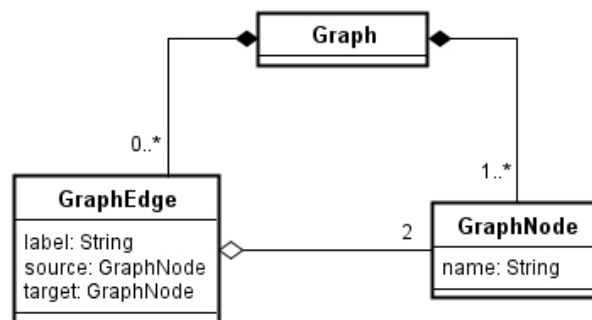


FIGURE 6.4 : diagramme UML de classe du graphe de situations.

Ainsi, chaque arête est un agrégat de deux nœuds. Le fait qu'il s'agisse d'un graphe orienté nécessite de préciser lequel de ces nœuds est la source de l'arc et lequel est la cible. Le libellé attaché à l'arête est indiqué par la chaîne de caractère *label*. De la même façon, à chaque nœud est associé une chaîne de caractère indiquant son nom.

Objets soutenant la méthodologie

Dans la méthodologie SAMANTA, les tâches de l'arbre sont liées aux nœuds du graphe de situation par la définition de contextes. Ceci se traduit par l'introduction d'objets supplémentaires dans le diagramme, présenté sur la figure 6.5 :

TaggedNode : L'objet *TaggedNode* est un objet intermédiaire permettant de définir une liste de tags associée à un nœud *GraphNode* donné afin d'être utilisé pour les méthodes de comparaison des tags. Si aucun tag n'est défini, l'objet *TaggedNode* ne contient qu'une référence vers le nœud original qu'il représente (et une liste vide).

ListNode : L'objet *ListNode* est utilisé afin de définir des listes de nœuds du graphe global, délimitant ainsi des sous-graphes à ce graphe global. Le fait que tous les nœuds appartiennent au même graphe global nous assure que les arêtes reliant les nœuds de la liste *ListNode* appartiennent également au graphe global. Les éléments de *ListNode* sont des nœuds tagués *TaggedNode*.

Cette structure est utilisée pour la construction de l'objet *Context*, et doit, au minimum, faire référence à un nœud du graphe.

Context : Cet objet permet de lier les tâches de l'arbre avec les sous-graphes du graphe de situation, par l'intermédiaire d'une liste *ListNode*. Ainsi, pour chaque tâche de l'arbre est défini un contexte *Context* auquel est associé le sous-graphe *associatedNodes* définissant les exigences en termes de SA.

La liste des nœuds saillants (tels que définis par l'utilisateur) est également associée à chaque instance de l'objet *Context* par l'intermédiaire d'un attribut *salientNodes* (également de type *ListNode*). Deux sous-graphes co-existent ainsi lors de la définition d'un contexte : les nœuds du sous-graphe *salientNodes* appartiennent nécessairement au sous-graphe *associatedNodes*.

TransitionAnalysis : Ce dernier objet n'est pas directement lié aux différents objets introduits jusqu'à présent, car il ne participe pas à la construction du modèle de données de l'application. En effet, celui-ci s'appuie sur les différentes données modélisées, en particulier les transitions et les contextes définis, afin de produire différentes analyses. Dans la version courante de l'application, trois méthodes d'analyse ont été développées :

- L'analyse des statistiques générales de la transition, notamment la comparaison des caractéristiques des tâches la composant (section 5.3.2). Il s'agit de la méthode *computeStats()* de l'objet *TransitionAnalysis*.
- L'extraction des plus grands sous-graphes communs (section 5.3.3), traduite par la méthode *extractLCSG()* de l'objet *TransitionAnalysis*.
- L'extraction des éléments saillants, selon la définition de Stanton, et tels que définis par l'utilisateur (modélisés par la liste *salientNodes*). Cette analyse (section 5.3.4) est exprimée par la méthode *extractSalientNodes()*.

Notons que l'objet *TransitionAnalysis* a été construit de façon à permettre l'ajout de nouvelles méthodes, et notamment celles décrites dans la section 5.3 de ce mémoire.

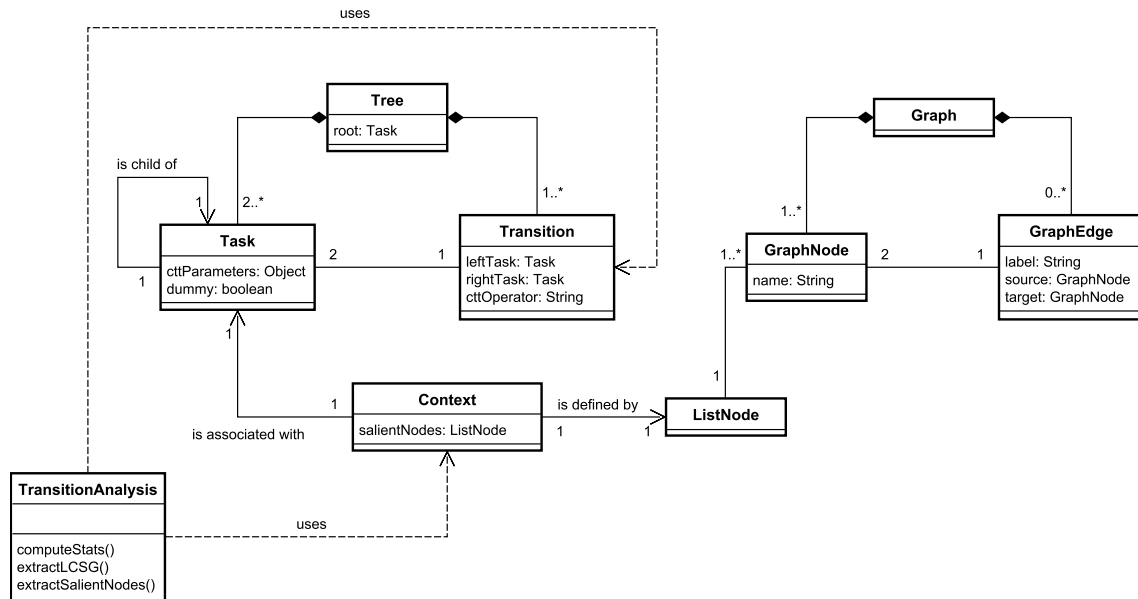


FIGURE 6.5 : les modèles manipulés par l'outil.

Ainsi, l'objet *TransitionAnalysis* s'appuie sur des objets *methodAnalysis* partageant une structure « de base » décrite dans le listing 6.1.

Listing 6.1: structure minimale d'un objet d'analyse de transition pour l'outil.

```

var methodAnalysis = {
  init: function(transition, name, idTab),
  buildContent: function(),
  destroyContent: function()
}
  
```

Pour chaque nouvel outil d'analyse, 3 méthodes doivent être implémentées :

- La méthode *init()* « instancie » l'objet avec les valeurs d'analyse courante (paramètre *transition*, contenant notamment les références aux tâches étudiées), le nom de la méthode d'analyse (qui sera le titre de l'onglet) et l'identifiant unique pour cette méthode (utilisé notamment dans la gestion de l'architecture DOM de la page).
- La méthode *buildContent()* est appelée lorsque l'onglet est affiché pour la première fois ou si les contextes associés à l'une des tâches de la transition ont été modifiés (afin de mettre à jour le contenu) ;
- La méthode *destroyContent()* est appelée lorsque les contextes associés à la tâche ont été modifiés et permet, en supprimant des objets de la mémoire notamment, de forcer le rafraîchissement du contenu de l'analyse.

Ainsi, l'ajout d'une nouvelle méthode d'analyse *methodAnalyse* des transitions peut se faire en créant un nouvel objet sur ce modèle et en le déclarant au sein de l'objet *TransitionAnalysis* qui s'occupe de transmettre les données issues des modèles et de gérer l'affichage des résultats de cette méthode d'analyse d'une transition (à travers

la création d'onglets dans une fenêtre modale, décrite dans la section 6.4.2.4).

6.3.3.3 Construction de l'application

L'application est construite à partir de plusieurs bibliothèques, décrites dans l'annexe C.1. Ces bibliothèques permettent de gérer à la fois l'affichage des éléments d'interface « classiques » (boutons, menus, onglets...) et de construire et afficher les modèles de tâche et de SA.

6.4 DESCRIPTION DE L'OUTIL

Dans cette section, nous décrivons succinctement l'outil développé dans le cadre de cette thèse. L'ensemble des interactions possibles et éléments d'interface ne sera pas décrit de manière exhaustive, la description sera focalisée sur les principales fonctionnalités permettant à l'utilisateur de réaliser les étapes 3 et 4 de la méthodologie.

L'outil développé fonctionne à la fois sur les ordinateurs et les tablettes (figure 6.6). La liste des navigateurs compatibles est présentée en annexe C.2.

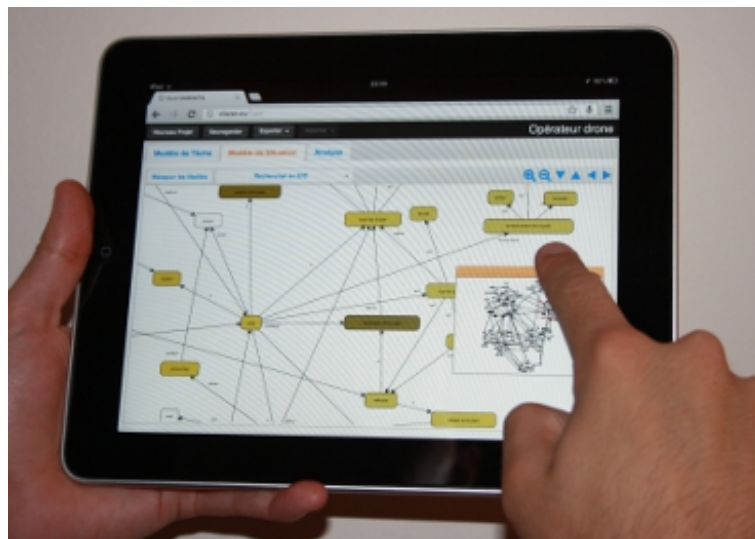


FIGURE 6.6 : photo de l'utilisation de l'outil sur une tablette tactile iPad.

6.4.1 Zones de l'écran

La figure 6.7 présentent la division de l'écran en trois zones principales. Ces trois zones correspondent aux trois fonctions suivantes :

1. La barre en haut de l'écran constitue la barre de *gestion de projet*, elle regroupe les boutons, menus et champs permettant à l'utilisateur de gérer le projet (création, sauvegarde, importation/exportation des données) et présente le titre du projet en haut à droite, modifiable par un double-clic (ou double tap sur les interfaces tactiles).
2. La barre d'onglet, située sous la barre de gestion de projet, permet de naviguer entre les deux modèles : le modèle de tâches et le modèle de situation, et également d'accéder aux fonctionnalités liées à l'analyse des transitions.
3. Enfin, la zone centrale présente le contenu des onglets de la barre n°2. C'est avec cette zone que l'utilisateur interagit principalement.

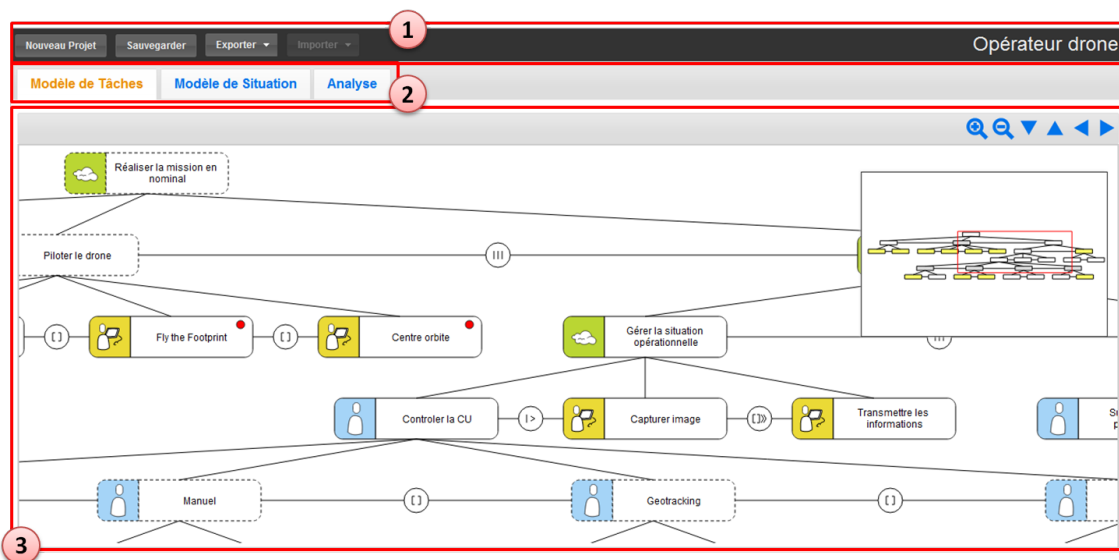


FIGURE 6.7 : capture d'écran des trois zones principales de l'outil SAMANTA.

Note sur la taille de l'application

La taille de l'application présentée est adaptée automatiquement à la taille et la résolution de l'écran de l'utilisateur. Ainsi, toutes les informations sont directement présentées à l'utilisateur-ergonome, qui n'a pas besoin de faire défiler l'écran pour accéder à d'autres menus ou fonctionnalités.

6.4.2 Fonctions de l'outil

La mise en œuvre de la méthodologie SAMANTA à travers l’outil associé se fait dans la continuité de l’utilisation des outils de modélisation appuyant la construction des modèles de tâche et de situation, en permettant l’important des fichiers résultant de leur utilisation. La suite de cette section décrit l’intégration à l’outil des étapes 3

et 4 de la méthodologie SAMANTA, à partir de l'importation des modèles jusqu'à l'analyse des transitions et l'export éventuel des données produites.

6.4.2.1 Importation des modèles

L'importation des deux modèles utilisés dans la méthodologie SAMANTA peut se faire de plusieurs manières, selon les données dont disposent l'utilisateur. Celle-ci se fait à travers le menu « Importer » de la barre de gestion de projet (figure 6.8).



FIGURE 6.8 : capture d'écran des options d'importation de données.

Trois possibilités sont présentées à l'utilisateur :

- L'importation de l'arbre de tâche uniquement, sous la forme d'un fichier XML issu de l'outil CTTE (annexe C.3.1 pour la description du format de fichier).
- L'importation de graphe de situation uniquement, sous la forme d'un fichier GraphML issu l'outil *yEd Graph Editor* (annexe C.3.2 pour la description du format de fichier).
- L'importation d'un fichier spécifique au format JSON, décrivant un projet entier, et issu d'un export précédent (annexe C.4.2 pour la description du format de fichier et le paragraphe 6.4.2.5 pour la description de la fonctionnalité d'exportation).

Lors de l'importation des fichiers, les modèles correspondant sont automatiquement construits et affichés par l'outil et sont accessibles à l'utilisateur à travers la barre d'onglet (zone n°2, figure 6.7). Si un de deux modèles a déjà été importé, l'option associée est grisée dans le menu.

6.4.2.2 Navigation et interaction avec l'arbre des tâches

L'affichage de l'arbre des tâches chargé par l'utilisateur se fait dans la zone centrale lorsque l'onglet « Modèle de tâches » est sélectionné. L'affichage des tâches et des transitions reprend les conventions établies par la notation ConcurTaskTree (Paternò *et al.*, 1997).

Nous utilisons l'algorithme de positionnement des nœuds d'un arbre défini par Walker II (1989), celui-ci permettant d'assurer un bon compromis entre lisibilité de l'arbre et compacité de l'ensemble (voir l'annexe C.1.2 décrivant les bibliothèques utilisées).

La figure 6.9 présente les différents éléments de la vue de l'arbre.

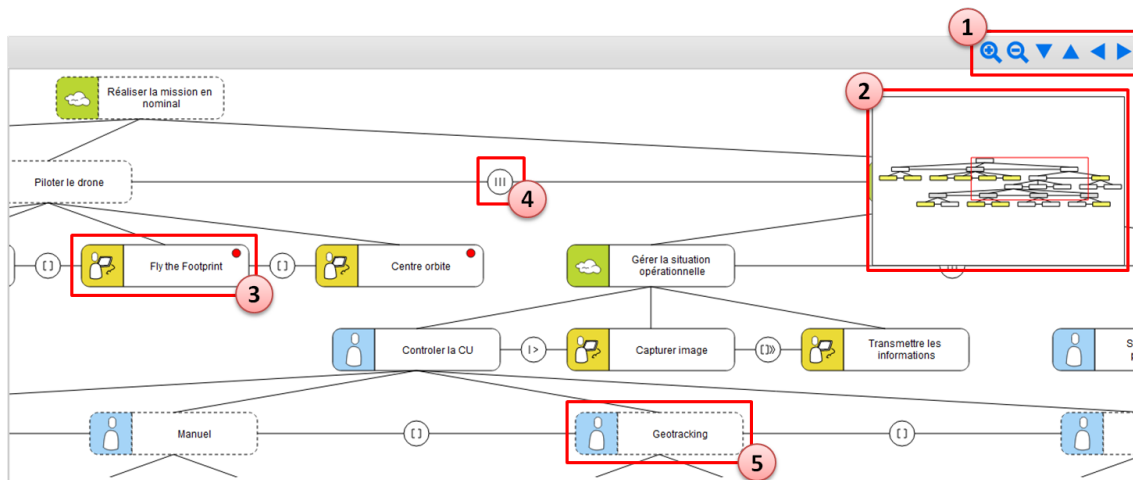


FIGURE 6.9 : capture d'écran de la vue « arbre de tâches ».

Navigation dans la vue

Plusieurs moyens de navigation ont été mis en place afin de permettre à l'utilisateur de se déplacer dans cette vue :

- Un groupe de boutons, présentés sous forme de flèches, est disponible afin de naviguer dans la vue par clic successifs (ou un clic continu), et permet ainsi à l'utilisateur de se déplacer « par palier » (zone 1, figure 6.9).
- Une vue miniature de l'ensemble de l'arbre est également disponible (zone 2, figure 6.9). Un cadre rouge représente la position actuelle de la vue centrale sur l'arbre et l'utilisateur peut interagir directement avec ce cadre, en cliquant au sein de la miniature pour le repositionner et ainsi recentrer la vue principale sur la zone correspondante.

La position de la miniature sur la vue principale peut être modifiée par l'utilisateur en cliquant sur un bandeau interactif (visualisé sur la figure 6.10) puis en « glissant-déplaçant » le cadre de la miniature à la position voulue. Ce bandeau apparaît au survol du curseur de la souris ou est affiché par défaut dans le cas d'interfaces tactiles (où la souris n'est pas présente).

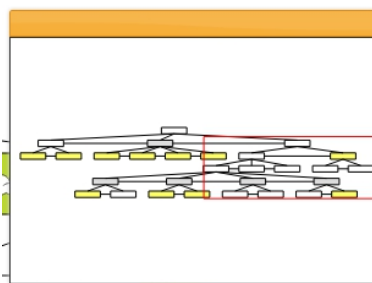


FIGURE 6.10 : capture d'écran de la miniature lors d'un survol souris. Une zone interactive apparaît afin de permettre à l'utilisateur de repositionner la miniature.

- Enfin, l'utilisateur peut se déplacer en utilisant le principe du « glisser-déplacer » directement sur la vue principale. Ainsi, il suffit à l'utilisateur de cliquer sur la vue principale, puis de déplacer celle-ci à la position voulue en maintenant le clic enfoncé.

Un système d'agrandissement / rétrécissement de la vue a également été développée afin de permettre à l'utilisateur de « zoomer » ou « dézoomer » sur certaines portions de l'arbre. Ces actions peuvent être effectuées :

- en utilisant les boutons correspondant de la zone 1 (figure 6.9), visualisé par des pictogrammes représentant une loupe avec un signe « plus » ou un signe « moins » ;
- en utilisant la molette de la souris, chaque sens de rotation correspondant à un agrandissement ou rétrécissement d'un palier de la vue principale. Dans le cas d'un fonctionnement sur tablette tactile, cette interaction a été remplacée par une interaction de type « pincement » ou « écartement », effectuée avec deux doigts.

Affichage des détails d'une tâche

La vue « arbre des tâches » permet de visualiser les différentes informations au sujet des tâches et transitions de l'arbre.

- La catégorie CTT de la tâche (parmi *abstrait*, *utilisateur*, *système* et *interactif*) est visualisée directement sous la forme d'un pictogramme sur fond coloré, situé à gauche de chaque tâche. Par exemple, la tâche encadrée par la zone 3 sur la figure 6.9 est une tâche interactive.
- Les tâches définies comme « factices » (voir la définition de ces tâches dans le paragraphe 5.2.2.2) sont visualisées par des bords pointillés (tâche encadrée par la zone 5 sur la figure 6.9).
- Les transitions sont représentées par les identificateurs CTT entre deux tâches (zone 4, figure 6.9). Un clic sur l'une d'entre elles affiche une fenêtre modale résumant les quelques détails de cette transition. Un double-clic permet d'accéder directement aux outils d'analyse de cette transition spécifique (voir aussi 6.4.2.4).

Un clic sur un bloc représentant une tâche permet d'afficher une fenêtre modale où sont résumées les informations disponibles sur cette tâche (figure 6.11). Au sein de cette fenêtre, l'utilisateur peut modifier la description de la tâche, voir quelles sont les tâches pouvant être exécutées en parallèles, ainsi que les éléments associés le cas échéant. La tâche peut également être définie comme factice directement à travers cette fenêtre.

Lorsqu'un contexte est défini pour la tâche, il est affiché à travers la coloration des nœuds d'un graphe miniature (à droite de la fenêtre modale). L'accès à ce contexte dans la vue « graphe de situation » se fait en cliquant sur le bouton correspondant de la fenêtre modale.

Notons qu'un double-clic sur l'une des tâches de l'arbre lorsqu'aucune fenêtre modale n'est ouverte produit le même effet et permet d'accéder directement au

contexte correspondant à cette tâche.

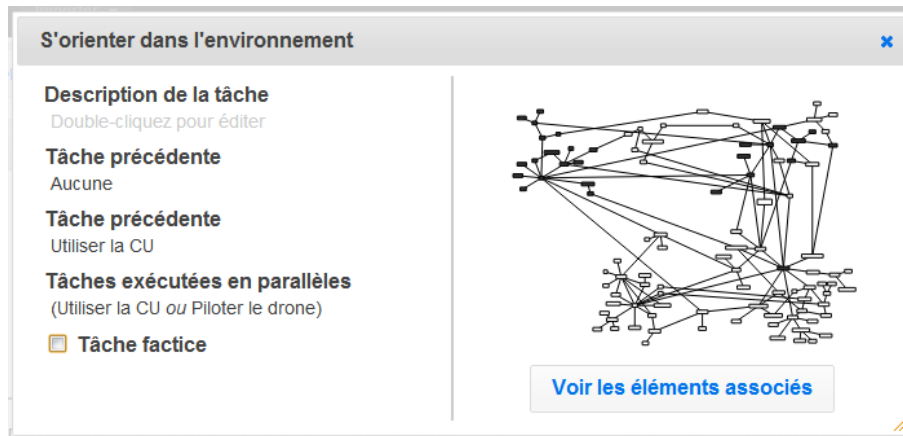


FIGURE 6.11 : capture d'écran de la fenêtre modale de détail d'une tâche.

6.4.2.3 Navigation et interaction avec le graphe de situation

La vue « graphe de situation » représente le contenu principal de l'onglet « Modèle de Situation ». Elle présente le réseau propositionnel des éléments de situation, défini par l'utilisateur en amont et importé dans le projet.

Les mêmes principes de navigation que ceux de la vue « Modèle de Tâches » ont été implémentés (zone 1, figure 6.12). Afin d'améliorer la lisibilité du graphe lorsque trop de libellés sont superposés, une option d'affichage/masquage de ces libellés est disponible (zone 2, figure 6.12).

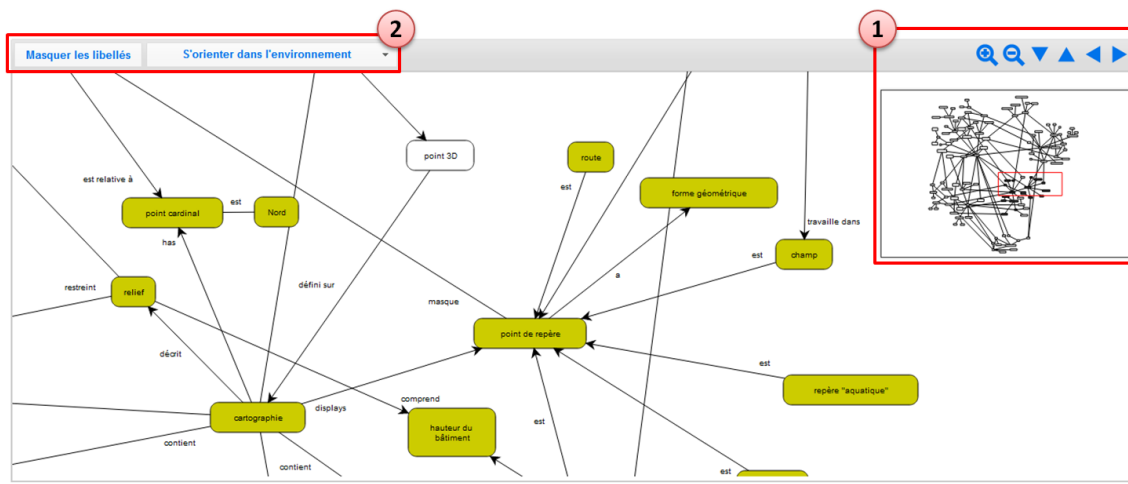


FIGURE 6.12 : capture d'écran de la vue « graphe de *Situation Awareness* ».

La fonction principale de cette vue est de permettre à l'utilisateur de visualiser les contextes associés à chaque tâche de l'arbre. Cette association est visualisée par

la coloration des nœuds du graphe correspondant, pour une tâche donnée. L'association ou la suppression d'un élément du contexte courant se fait par un double-clic sur cet élément.

Sélection de la tâche courante

Le menu déroulant de la zone 2 de la figure 6.12 permet de visualiser le titre de la tâche courante, pour laquelle l'analyste est en train de définir un contexte. Cette tâche peut être changée en sélectionnant une autre tâche de l'arbre directement dans ce menu (figure 6.13). Au survol des options du menu, la vue principale de l'onglet « Modèle de Situation » présente un aperçu des éléments associés à la tâche survolée. Si aucun élément n'est associé, alors les éléments « hérités » seront affichés dans une couleur différentes, mais disparaîtront à la sélection de la tâche concernée, affichant un graphe vierge où l'analyste pourra définir un « vrai » contexte.

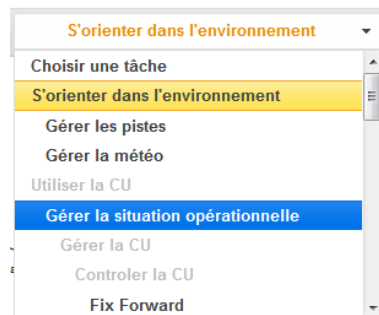


FIGURE 6.13 : capture d'écran du menu déroulant de sélection de la tâche courante dans l'onglet « Modèle de Situation ».

Notons que la sélection de la tâche courante pour la définition de contexte peut également se faire à partir de l'arbre des tâches (§ 6.4.2.2)

Affiche des détails d'un élément de situation

L'utilisateur peut accéder aux informations concernant un élément de situation particulier en cliquant sur le nœud correspondant du graphe de situation. La fenêtre modale affichée (figure 6.14) permet ainsi de voir l'ensemble des tâches auxquelles l'élément est associé (à droite de la fenêtre) ainsi que les autres nœuds liés à cet élément et les tags associés pour la tâche courante (à gauche).

Le caractère *saillant* de l'élément pour l'une ou l'autre des tâches auxquelles il est associé peut être défini à l'aide de l'étoile située à gauche du titre de chaque tâche. Dans l'exemple de la figure 6.14, l'élément « *véhicule* » est défini comme saillant par l'utilisateur pour la tâche « Surveiller cible en IR ». De plus, pour la tâche « Rechercher en E/O », qui est la tâche courante, le tag « mobile » a été défini.



FIGURE 6.14 : capture d'écran de la fenêtre modale de détails d'un élément de situation.

6.4.2.4 Analyse des transitions

Suite à l'étape de définition des contextes associés à chaque tâche, appuyée par le contenu des deux premiers onglets de l'outil, vient l'étape d'analyse des transitions, entièrement appuyée par le troisième onglet, intitulé « Analyse » (figure 6.15). Cet onglet présente l'ensemble des analyses déjà créées pour le projet courant et permet d'ajouter de nouvelles analyses.

Les analyses déjà créées sont visualisées sous forme de blocs (numéro 2 de la figure 6.15 par exemple). Au survol de l'un de ces blocs, les deux tâches correspondantes sont mises en exergue sur une miniature (cadre 1, figure 6.15). Comme pour la miniature des deux autres onglets, celle-ci peut être déplacée par « glisser-déposer ».

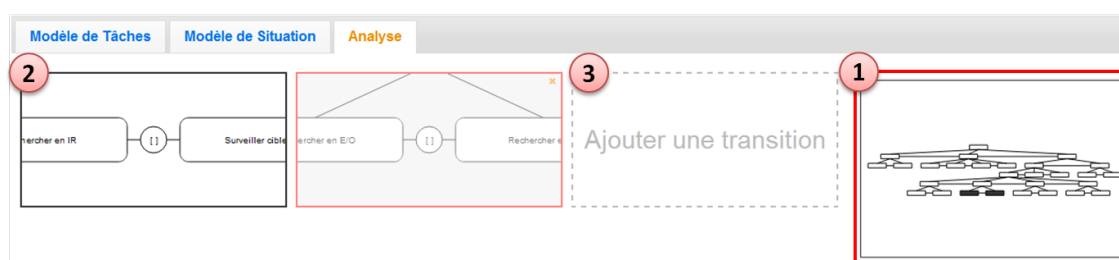


FIGURE 6.15 : capture d'écran de la vue centrale de l'onglet « Analyse ».

Ajout d'une analyse de transition

L'ajout d'une nouvelle analyse peut se faire de deux façons différentes :

- Pour les transitions directement accessibles sur l'arbre de tâches, un double-clic sur le cercle correspond de la vue « Modèle de Tâches » ajoute la transition à l'onglet « Analyse ».
- Pour toutes les transitions, l'utilisateur peut cliquer sur le bloc « Ajouter une transition » (numéroté 3 sur la figure 6.15). Une fenêtre modale de sélection

des tâches est alors affichée (figure 6.16). L'utilisateur doit alors cliquer successivement sur les deux tâches composant la transition à analyser, l'outil se chargeant de définir l'identificateur CTT les liant, conformément à l'algorithme proposé dans le chapitre précédent (section 5.3.1.1). La transition est visualisée par une ligne rouge sur la figure 6.16.

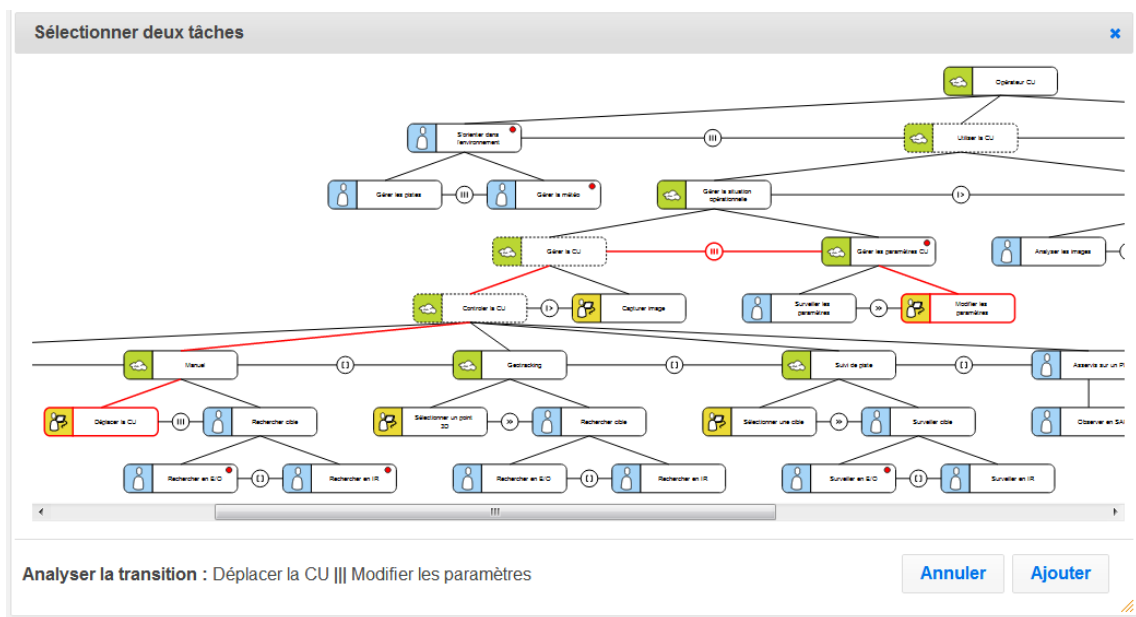


FIGURE 6.16 : capture d'écran de la fenêtre modale de sélection d'une transition.

Méthodes d'analyse ajoutées

Parmi les différentes méthodes présentées dans la section 5.3, trois outils d'analyse ont jusqu'à présent été implémentés au sein de l'outil. Ces différentes méthodes sont présentées dans une même fenêtre modale, affichée lors d'un clic sur le bloc de la transition. Chaque méthode est visualisée à travers la sélection d'un onglet.

1. Analyse des statistiques

Ce premier onglet, illustré figure 6.17, présente les statistiques générales relatives aux deux tâches impliquées dans la transition. L'utilisateur peut ainsi voir un résumé des éléments propres à chaque tâche ainsi que les éléments partagés. Une mesure du score de Tversky est également affichée, ainsi que la classification de la transition dans la taxinomie des transitions.

Il est également donné la possibilité à l'utilisateur d'inclure les tâches parallèles dans l'analyse, ce qui aura pour effet de rafraîchir la vue.

2. Extraction des plus grands sous-graphes communs

La méthode d'extraction des plus grands sous-graphes communs, présentée dans la section 5.3.3 est intégrée dans le deuxième onglet de cette fenêtre

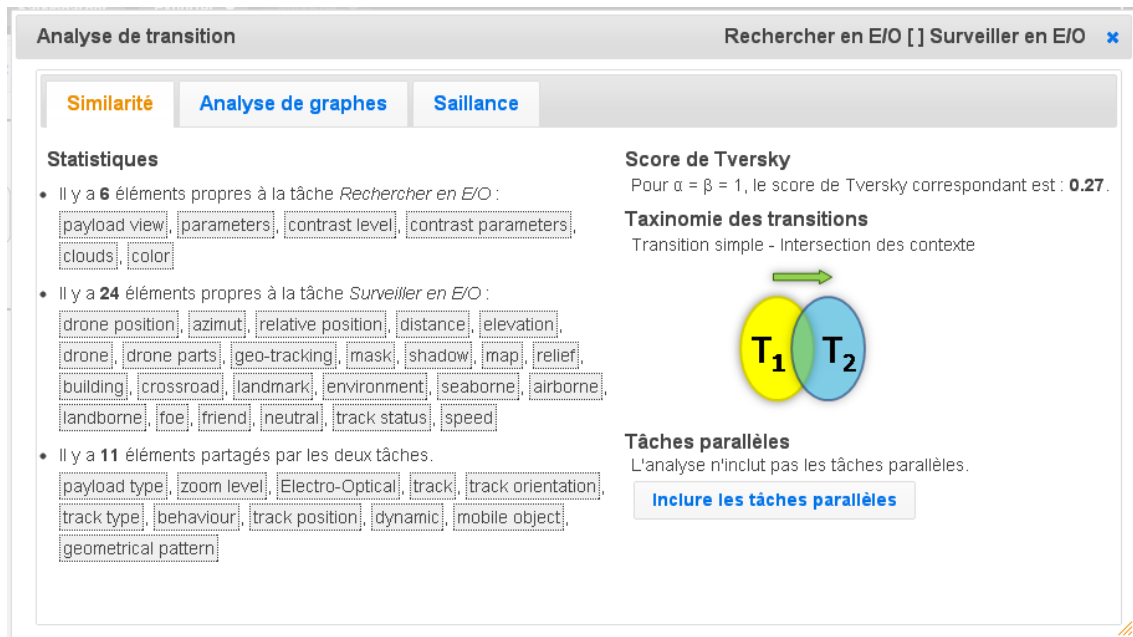


FIGURE 6.17 : capture d'écran de la fenêtre modale d'analyse d'une transition, présentant les statistiques générales relatives à la transition.

d'analyse. Cet onglet permet de naviguer à travers les différents sous-graphes communs aux deux tâches, en intégrant le cas échéant les tâches parallèles. La figure 6.18 présente la disposition de cet outil d'analyse.

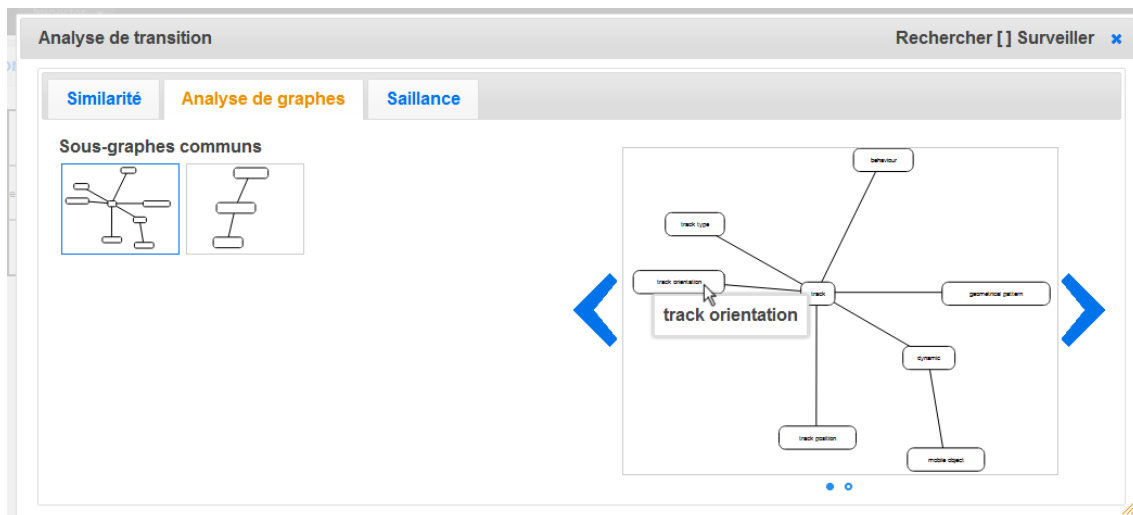


FIGURE 6.18 : capture d'écran de la fenêtre modale d'analyse d'une transition, présentant l'outil d'extraction des plus grands sous-graphes communs.

3. Recherche des éléments saillants

Les deux méthodes de recherche des éléments saillants, à savoir ceux définis par l'utilisateur et ceux répondant à la définition de Stanton (section 5.3.4), sont visualisées au sein du même onglet (figure 6.19).

La partie gauche de la vue affiche les éléments saillants, selon les deux métriques, et pour chacune des deux tâches en jeu tandis que la partie droite présente une vue miniaturisée du graphe de situation. L'utilisateur peut visualiser sur cette miniature les nœuds des différents ensemble de la liste de gauche, en survolant les groupes à la souris ou en survolant chaque nœud individuellement (ou par un *tap* sur les interfaces tactiles).

Par exemple, sur la figure 6.19, les trois nœuds définis comme saillants pour la tâche « Rechercher en IR » par l'utilisateur sont mis en exergue sur la miniature par des cercles rouges.

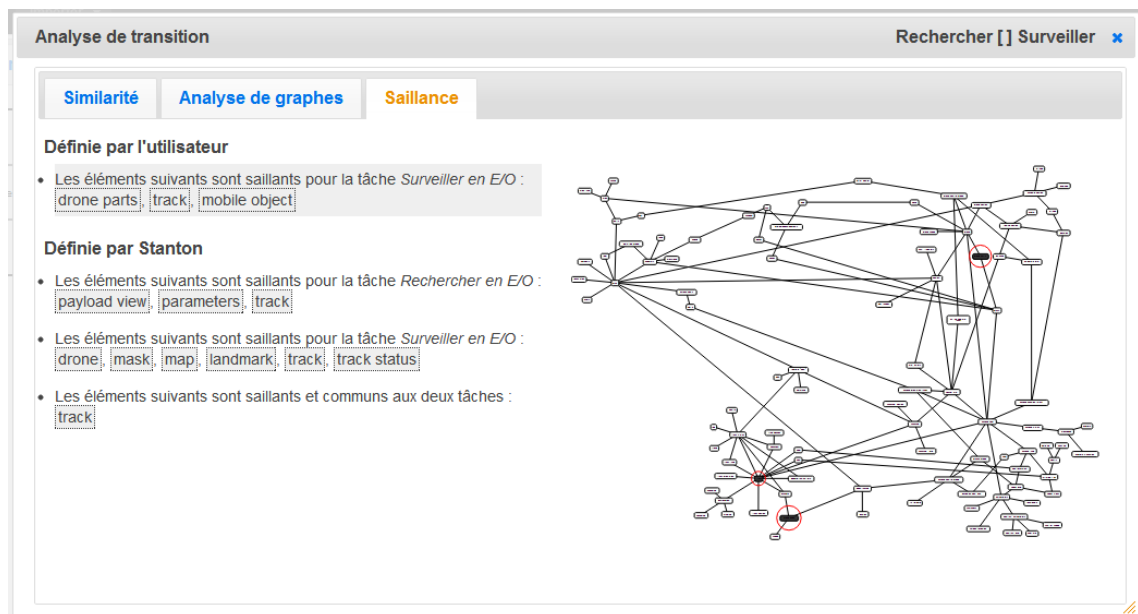


FIGURE 6.19 : capture d'écran de la fenêtre modale d'analyse d'une transition, présentant l'outil d'analyse de la saillance.

6.4.2.5 Sauvegarde et exportation des données

Un processus de sauvegarde automatique a été mis en place au sein de l'outil, s'appuyant sur la technologie *localStorage* de HTML5 et permettant de stocker dans une base de données locale au navigateur de l'utilisateur les différents modèles importés ainsi que les modifications effectuées (définition des contextes associés aux tâches, liste des transitions analysées, nom du projet...). Cette sauvegarde est déclenchée toutes les cinq minutes de manière automatique, et peut être déclenchée

manuellement par l'utilisateur grâce au bouton « Sauvegarder » de la barre de gestion de projet. Ainsi, lorsque l'utilisateur quitte le navigateur, si celui-ci est compatible⁶ avec cette technologie, les données sont enregistrées localement pour être retrouvées lors de l'utilisation suivante (pouvant s'effectuer hors ligne).

Un système d'export des données au format JSON (annexe C) a également été mis en place, permettant à l'utilisateur d'enregistrer l'ensemble des données relatives à un projet sous la forme d'un fichier unique, pouvant être partagés avec d'autres utilisateurs et importés par eux sur d'autres supports utilisant le même outil.

Enfin, un moyen d'exportation des graphes sous forme d'image (au format PNG) a été créé afin de pouvoir partager, de manière visuelle, les contextes associés à chaque tâche. Nous utilisons notamment cette fonction au sein de ce mémoire afin de produire les figures du chapitre suivant.

6.5 PERSPECTIVES D'ÉVOLUTION DE L'OUTIL

L'objectif fixé de développement d'un outil prenant en charge les étapes 3 et 4 de la méthodologie SAMANTA est rempli par l'application présentée dans la section précédente. Néanmoins, dans la perspective d'évolutions futures de cet outil, nous avons identifié plusieurs développements et ajouts dont l'exécution permettrait d'enrichir l'utilisation de cet outil.

6

6.5.1 Mise en place d'autres méthodes d'analyse des transitions

Le chapitre précédent fait état d'autres méthodes d'analyse que nous n'avons pas développées dans cette première version de l'outil. Ainsi, deux méthodes supplémentaires de comparaison des tâches impliquées dans une transition pourront être ajoutées : l'analyse de la frontière (§ 5.3.2.3) et l'analyse des métadonnées associées à la transition (section 5.3.5). Ce développement est facilité par la mise en place d'une structure commune pour le développement des différentes méthodes d'analyse (voir § 6.3.3.2 pour une description de cette structure).

6.5.2 Identification des transitions difficiles

Dans la section 4.3.1, nous avons présenté nos réflexions concernant l'analyse prédictive des transitions de l'arbre de tâche. Parmi les suggestions proposées, la mesure du score de Tversky et la classification des transitions en fonction de ce score est un processus pouvant être automatisé au sein d'un outil tel que celui que nous avons développé.

Ainsi, nous pouvons imaginer que, dans le cas où les associations entre contextes et tâches ont été définies pour l'ensemble des tâches, une fonction permettant à

6. voir aussi annexe C.2 pour la liste des navigateurs compatibles

l'utilisateur-ergonome de lancer un traitement automatisé des données qu'il a entrées dans l'application pourrait être ajoutée. Celle-ci aurait pour objectif de détecter et de mettre en valeur les transitions que jugées « difficiles » relativement les unes aux autres en fonction du score de Tversky et de la classification taxinomique proposée.

Cette analyse automatisée pourrait venir en aide à l'analyste afin de détecter les transitions sur lesquelles il doit focaliser son analyse à l'aide des outils et méthodes existants. Sa mise en œuvre nécessite toutefois d'évaluer au préalable les impacts potentiels des transitions sur l'opérateur en fonction de leur catégorie dans la taxinomie.

6.5.3 Intégration d'autres outils à la méthodologie

L'outil développé et présenté dans ce chapitre s'intègre dans une logique plus globale « d'outillage » de la méthodologie, en s'appuyant principalement sur les deux outils de créations des modèles (CTTE pour l'arbre de tâche et *yEd Graph Editor* pour le graphe de situation).

Afin de compléter cette « chaîne » d'outils et de soutenir les étapes 1 et 5 de la méthodologie à l'aide d'outils pouvant simplifier le travail de l'analyste, nous proposons quelques pistes illustrées d'outils pouvant s'intégrer à cette logique.

6.5.3.1 Extraction des connaissances

L'extraction des connaissances à partir des transcriptions d'entretiens ou de la littérature est un travail fastidieux, dont la difficulté augmente proportionnellement au nombre de sources de données à analyser, regrouper et compiler. Des outils permettent de soutenir l'analyste dans sa tâche de traitement des données.

L'outil REMINER de Meth *et al.* (2012) permet par exemple de détecter de manière semi-automatique les tâches de l'utilisateur en analysant des textes rédigés dans le langage naturel, combinant le traitement du langage naturel et l'annotation des textes par l'analyste pour extraire ces tâches. Cet outil est spécifiquement destiné à l'analyse des tâches de l'utilisateur et peut donc permettre d'aider l'analyste à construire l'arbre des tâches.

A cette fin, l'outil CTTE (Mori *et al.*, 2002) inclut une fonctionnalité d'aide à l'analyse des scénarios et à leur traduction en rôles, objets et tâches (outil « *Informal to formal description* »). Cette fonctionnalité de l'outil CTTE n'est pas automatisée mais peut aider l'analyste à formaliser l'arbre des tâches de l'opérateur, en ayant l'avantage d'être intégrée directement à l'outil utilisé pour cette formalisation réduisant ainsi le nombre d'étapes entre le traitement et la modélisation.

Afin d'appuyer l'analyste dans l'extraction des concepts des transcriptions d'entretiens pour leur intégration au réseau propositionnel de connaissances, de nombreux travaux ont été effectués dans le domaine du traitement automatique de la langue, en particulier pour notre cas d'application, dans le domaine de la reconnaissance d'entités nommées (classification des mots en fonction de leur catégorie : nom de personne, de lieu, distance, date...).

6.5.3.2 Recommandations IHM

Les recommandations émises par l'analyste à la suite de l'analyse des transitions peuvent être exprimées de plusieurs manières, en s'appuyant notamment sur des outils de maquettage et de prototypage d'interface. Ainsi, l'outil *Balsamiq Mockup*⁷ permet de créer des maquettes d'interfaces en s'abstrayant de l'aspect graphique, la visualisation des différents éléments d'interface se faisant à travers un effet « fil de fer ». Cette première étape de maquettage peut être renforcée par un prototype interactif, construit par exemple par l'application *Axure RP*⁸, pouvant servir de base pour des tests utilisateurs.

6.6 SYNTHÈSE DU CHAPITRE 6

Dans ce chapitre, nous avons présenté la démarche mise en œuvre pour la conception d'un outil à destination des ergonomes et experts facteurs humains, dans le but de simplifier la mise en œuvre de la méthodologie SAMANTA.

Cet outil, développé sous la forme d'une application web dynamique, a été utilisé afin d'analyser les transitions de trois cas d'étude :

- l'illustration de la méthodologie SAMANTA, présentée en conclusion du chapitre précédent (section 5.5) ;
- une application à la conception d'un système de commande/contrôle de drone, présenté dans le chapitre suivant ;
- une application au cas de la conduite des centrales nucléaires, en utilisant le cas du réacteur EPR (section 7.8). Cette application, présentée succinctement dans le cadre de ce mémoire, a été réalisée par une ergonome experte dans le domaine et fait l'objet d'une publication (Villaren *et al.*, 2012b).

7. disponible sur <http://www.balsamiq.com/>

8. disponible sur <http://www.axure.com>

7

Opérationnalisation de la méthodologie

Dans ce chapitre, nous présentons l'application de la méthodologie SAMANTA aux systèmes de drone, à travers l'étude d'un scénario opérationnel issu d'entretiens avec les opérateurs de drones en exercice au sein de l'armée française. Ce cas d'application s'appuie sur un ensemble de données collectées au cours d'une étude menée par Bertin Technologies (présentée en annexe A), et financée par la Direction Générale de l'Armement (DGA).

La deuxième partie de ce chapitre présente la mise en œuvre de la méthodologie SAMANTA par une ergonome de Bertin Technologies et son application au cas de la conduite de centrale nucléaire.

7.1 LES SYSTÈMES DE DRONE

Un drone (ou UAV en anglais, pour « *Unmanned Aerial Vehicle* ») peut être défini comme un avion sans équipage, utilisé pour des missions diverses. Il existe une multitude de drones, de dimensions et de fonctions différentes, à voilure tournante ou fixe. Leur pilotage peut se faire de manière automatique et à distance, à partir d'un poste de commande au sol. Les drones emportent au moins une charge utile, pouvant servir à la reconnaissance ou surveillance de zone (caméra embarquée par exemple), ou au combat (missiles air-sol par exemple).

Les systèmes de drone considérés dans cette étude sont constitués de deux éléments :

- le drone (aussi appelé *vecteur*), appareil dont l'attitude¹ de vol caractérise la classe, embarquant une *charge utile* (caméra de surveillance ou arme pour les drones de combat) ;
- la station de contrôle/commande, aussi appelée « station sol ». Celle-ci peut être installée dans une zone proche du théâtre d'opération où interviennent les drones, comme c'est le cas dans les armées françaises, ou à de très grandes distances, comme pour certains systèmes américains tel que le MQ-1 Predator, opérés depuis l'état du Nevada, aux États-Unis (Gertler, 2012).

1. L'attitude est caractérisée par la vitesse du drone, son endurance et son altitude maximum.



(a) vue d'artiste.



(b) SDTI catapulté depuis son lanceur en opération.

FIGURE 7.1 : le drone SDTI construit par Sagem DS (Source : www.sagem-ds.com).

Deux systèmes de drone sont actuellement utilisés par les forces armées françaises : le système SDTI et le système Harfang. Dans leur configuration actuelle, ces systèmes remplissent des missions touchant à tous les aspects du renseignement (observation, détection, surveillance, identification) et interviennent dans de nombreuses missions d'appui à l'engagement, mais ne sont pas eux-mêmes armés.

Le drone tactique SDTI (Système de Drone Tactique Intérimaire, figure 7.1) construit par Sagem DS est utilisé par le 61^{ème} Régiment d'Artillerie de Chaulmont (Armée de Terre). Ce drone à aile delta se déplace à une vitesse située autour de 180 km/h, pour une altitude maximum de 3 800 mètres (ce plafond définit le caractère « tactique » du drone) et une autonomie d'environ quatre heures. Ce drone est lancé à partir d'une catapulte (figure 7.1b) et récupéré à l'aide d'un parachute.

L'équipe en charge du vol, au sein de la station sol, est composée :

- d'un opérateur « Vecteur » (le pilote) ;
- d'un opérateur « Charge Utile » (contrôlant la caméra embarquée dans la « boule » située sous le drone) ;
- d'un chef de mission supervisant l'équipe.

Certains entretiens avec des opérationnels ont indiqué qu'un analyste image était parfois présent directement au sein de la station sol, afin de permettre à l'opérateur CU de consacrer 100 % de son activité sur la tâche de surveillance / détection, prenant les captures et les transmettant à l'analyste.

Le drone Moyenne Altitude/Longue Endurance (MALE) Harfang (anciennement SIDM, Système Intérimaire de Drone MALE, figure 7.2) construit par Cassidian AS est utilisé par l'escadron Drone de Mont-de-Marsan (Armée de l'Air). D'une autonomie de 24 heures, ce drone vole à une altitude



(a) vue d'artiste (Source Cassidian AS). (b) au salon du Bourget 2009 (photo : Patrick Giraud).

FIGURE 7.2 : le drone Harfang construit par Cassidian AS.

de 7 500 mètres (qualifiée de « moyenne altitude ») pour une vitesse ne dépassant pas les 200 km/h. Drone à voilure fixe, son décollage et son atterrissage s'effectuent sur la même piste que les aéronefs « habités ».

L'équipe de la station sol est composée :

- d'un opérateur « pilote », en charge du vol et du contrôle de la caméra embarquée ;
- d'un analyste image qui étudie en temps réel les images transmises par le « pilote ». Il n'intervient pas directement sur le contrôle de la charge utile mais peut toutefois demander au pilote de viser un endroit particulier ou de repasser sur une zone ;
- d'un officier de renseignement définissant en temps réel les zones à étudier, selon les objectifs, le type de recherche, etc.

Des charges utiles supplémentaires peuvent être ajoutées (des radars par exemple), et sont contrôlées par des opérateurs supplémentaires.

7.2 MISE EN ŒUVRE DE LA MÉTHODOLOGIE SAMANTA DANS LES SYSTÈMES DE DRONE

Afin de nous placer dans une optique de conception d'un système de drone et de nous focaliser sur l'analyse des transitions d'un tel système, nous réduisons notre étude aux tâches d'un opérateur unique, réalisant les tâches de contrôle du vecteur et celles de renseignement (surveillance, détection, reconnaissance, identification) à travers l'usage de la charge utile.

Cette configuration correspond à celle actuellement mise en place dans les stations de contrôle du drone MALE Harfang de Cassidian AS au sein de l'Armée de l'Air : l'opérateur « principal » est tour à tour pilote puis gère la charge utile, lais-

sant les automatismes gérer la tâche de pilotage.

Afin de cadrer la conception de l'interface de l'opérateur en charge du pilotage et de la charge utile, nous faisons l'hypothèse que dans l'interface conçue, l'opérateur aura au moins accès à deux vues pouvant être affichées simultanément :

- une vue cartographique présentant les informations sur l'environnement de manière absolue, dans un système de coordonnées standard ;
- une vue présentant le flux vidéo transmis depuis la charge utile embarquée, offrant un point de vue relatif à l'opérateur. Bien que plusieurs types d'optiques peuvent être embarquées au sein de la charge utile (caméra « jour », infrarouge...), nous faisons l'hypothèse que la bande passante allouée au flux vidéo ne permet que la transmission d'un des flux à la fois, afin de conserver une qualité d'image optimale.

Les sous-sections suivantes présentent la mise en œuvre de la méthodologie SAMANTA à partir des données récoltées dans le cadre du projet FH/PA (présenté en annexe A).

7

7.3 EXTRACTION DES CONNAISSANCES

Les connaissances utilisées sont issues majoritairement des travaux de collecte de données effectués dans le cadre du PEA FH/PA. Les experts en Facteurs Humains de Bertin Technologies ont mené des entretiens auprès d'experts et d'opérationnels et ont analysé la littérature du domaine. A l'issue de ces entretiens des fiches décrivant les différents faits de RETEX (RETour d'EXpérience) ont été produites et des Situations d'Activité Caractéristiques (SAC) extraites.

7.3.1 Sources de données

Dans le cadre de l'étude amont FH/PA, 18 entretiens ont été réalisés à la fois auprès d'opérationnels des armées et d'experts industriels, ayant les profils suivants :

- opérateurs de drones tactiques (modèle SDTI, voir section 7.1) du 61^{ème} Régiment d'Artillerie de Chaumont (Armée de Terre) ;
- opérateurs de drones MALE (modèle Harfang, voir section 7.1) de l'escadron drone de Mont-de-Marsan (Armée de l'Air) ;
- experts drones du constructeur Sagem DS, ayant participé à la conception et/ou la formation des opérateurs du système SDTI ;
- experts drones du constructeur Cassidian AS, ayant participé à la conception et/ou la formation des opérateurs du système Harfang.

Les entretiens réalisés, de type semi-directif, s'appuyaient sur des guides d'entretien conçus par les experts en facteurs humains de Bertin Technologies. Ces guides intégraient les différents aspects Facteurs Humains, technologiques et organisationnels

jugés pertinents pour l'étude et identifiés au préalable dans la littérature et lors de sessions de travail avec les industriels « dronistes » du groupement.

Les industriels constructeurs de drones ont également mis à disposition du groupement des éléments issus de leurs bases de données, notamment des comptes rendus de vols en opération ou d'entraînement.

Enfin, diverses sources bibliographiques ont été explorées : les rapports publics du Bureau Enquêtes Accidents Défense Air (BEAD-Air), des publications scientifiques rapportant des incidents ou des particularités des systèmes, ainsi que des articles parus dans des revues spécialisées.

Ces entretiens et la veille bibliographique réalisée par le groupement ont également permis d'obtenir des données sur d'autres systèmes utilisés par des armées étrangères :

- le drone tactique Sperwer SPCAN construit par Sagem DS (modèle très proche du SDTI français), utilisé par les forces canadiennes ;
- le drone MALE MQ-1 Predator construit par *General Atomics Aeronautical Systems*, utilisé par l'armée américaine.

Dans le cadre de l'étude, d'autres systèmes dont les caractéristiques (missions, organisation, interfaces et allocation des tâches) sont connexes aux problématiques Facteurs Humains des systèmes de drone ont également été étudiés : la Patrouille Maritime (PATMAR), le contrôle aérien (ATC), la Circulation Aérienne Générale (CAG) et la Circulation Aérienne Militaire (CAM).

7

7.3.2 Exploitation des données collectées

Au total, près de 200 faits de RETEX, décrivant des faits explicites et concrets (positifs ou négatifs), ont été étudiés. Sur ces 200 faits, 40 ont été retenus pour leur pertinence et leur représentativité de l'ensemble des problématiques abordées : 10 concernaient les drones MALE, 20 les drones tactiques et les 10 derniers se rapportaient à des problématiques entre les systèmes de drone et les systèmes « connexes ».

Dans le cadre de l'application de la méthodologie SAMANTA, les données décrites dans les fiches de RETEX ont été utilisées afin de construire l'arbre des tâches et le modèle de *Situation Awareness* présentés dans la section suivante. Elles ont été complétées par des échanges avec les industriels experts drones et les opérateurs des armées rencontrés pendant les deux campagnes d'expérimentation du projet FH/PA (courant 2010).

7.4 MODÉLISATION

Dans notre cas d'application, nous considérons un scénario type issu d'un entretien réalisé auprès d'un opérateur drone de l'armée de l'air (drone MALE Harfang).

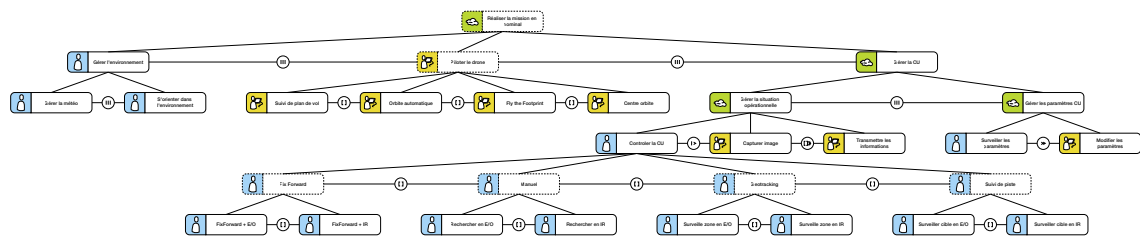


FIGURE 7.3 : vue d'ensemble du modèle de tâches.

Lors de cet entretien, l'opérateur a décrit les différentes phases d'une mission réalisée en mode nominal, c'est-à-dire sans panne du vecteur ni de la charge utile. Ainsi, dans les modèles présentés, nous n'intégrons pas la gestion des pannes ni les activités réalisées en amont et en aval de la mission, à savoir la préparation de mission, la vérification des paramètres de vol, la création de plans de vol, les éléments relatifs au décollage et l'atterrissage, ou encore les débriefings post-mission.

7

7.4.1 Modèle de tâches

La figure 7.3 présente le modèle de tâches dans sa totalité. Il est composé de trois tâches principales se déroulant en parallèle : la gestion des informations de l'environnement, le pilotage du drone et la gestion de la charge utile. Conformément à la relation de parallélisme définie par la notation CTT, ces tâches ne sont pas nécessairement actives 100 % du temps ni simultanément, mais elles peuvent se chevaucher lors de certaines phases de mission. Dans la suite, nous décrivons les sous-tâches associées à chacune des trois tâches principales.

7.4.1.1 Tâche 1 : gérer l'environnement

Nous avons regroupé au sein de cette tâche les différentes sous-tâches relatives à l'insertion du vecteur dans l'environnement. La figure 7.4 présente les deux sous-tâches de cette tâche principale.

Gérer la météo : la gestion de la météo consiste à tenir compte des différents aspects météorologiques lors des phases de la mission. Par exemple, le vent peut influencer la trajectoire du vecteur, le soleil peut produire des reflets sur la caméra Electro/Optique, les nuages peuvent s'interposer entre la charge utile et le sol, masquant les cibles potentielles, etc.

S'orienter dans l'environnement : cette tâche, primordiale pour l'opérateur, lui permet de se repérer tout au long de la mission, par rapport aux repères géographiques (bâtiments, cours d'eau...) ou aux objectifs de missions (points de passage, zones de reconnaissance...).

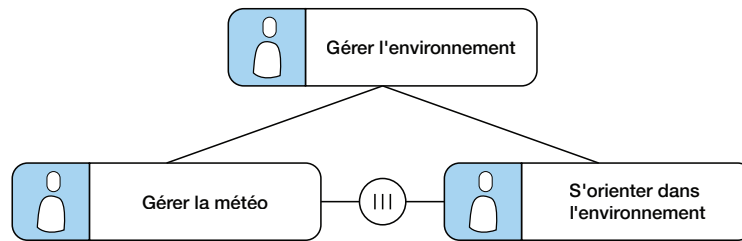


FIGURE 7.4 : la tâche « Gérer l'environnement » et ses sous-tâches.

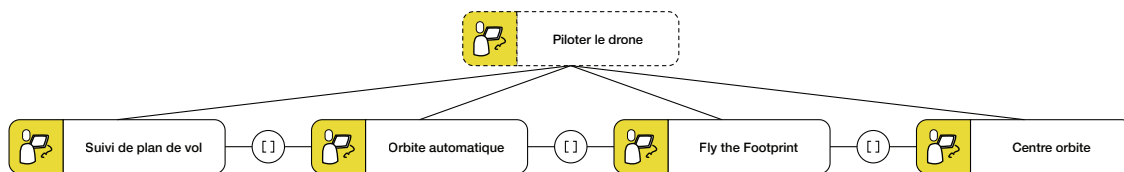


FIGURE 7.5 : la tâche « Piloter le drone » et ses sous-tâches.

7.4.1.2 Tâche 2 : piloter le drone

Comme nous l'avons décrit dans le scénario type, le pilotage du drone est une tâche continue pour l'opérateur. Dans certaines phases de mission, il s'agit de sa tâche principale, notamment lorsqu'il transite depuis la zone de décollage jusqu'à la zone de mission ; dans d'autres phases, cette tâche de pilotage est asservie au contrôle de la charge utile et l'exploration de la zone considérée. Dans notre cas d'application, nous considérons quatre modes de contrôles pour le pilotage du drone (figure 7.5) :

Suivi de plan de vol : le drone suit un plan de vol prédéfini ou créé en temps réel. Ce plan de vol est composé de points de passages (*waypoints*), desquels le drone se rapproche généralement au mieux en respectant les contraintes physiques qui lui sont imposées.

Orbite automatique : le drone effectue une orbite automatiquement autour du point visé par la charge utile de l'opérateur. Cette orbite est définie par l'inclinaison de la charge utile et la distance du drone par rapport au sol, que le système va tenter de conserver lorsque l'opérateur change de zone observée.

Fly the Footprint : le drone suit la trace au sol de la charge utile, c'est-à-dire qu'il se déplace dans la direction observée par l'opérateur avec la charge utile, lorsqu'il la contrôle en « mode manuel ». Il s'agit également d'un mode de pilotage asservi au contrôle de la charge utile.

Centre orbite : le drone effectue une orbite autour d'un point définissant son centre. Ce mode est par exemple utilisé dans le cadre de la surveillance d'objets particuliers (protection d'un bivouac, par exemple).

7.4.1.3 Tâche 3 : gérer la charge utile

La tâche principale de gestion de la charge utile (figure 7.6) est composée de plusieurs sous-tâches, certaines ayant été créées afin de lever les ambiguïtés entre les identificateurs CTT (tâches « factices »). Ainsi, la gestion de la CU est divisée en deux tâches parallèles (figure 7.6a) :

- La **gestion des paramètres de la charge utile** qui consiste à surveiller les paramètres relatifs aux optiques (mise au point, focale, contraste...). Ces paramètres sont généralement réglés avant la mission mais doivent être surveillés tout au long de celle-ci et peuvent nécessiter des modifications. C'est également à travers cette tâche que l'opérateur sera notifié d'une éventuelle panne de la charge utile, sans généralement pouvoir y remédier à distance.
- La **gestion de la situation opérationnelle** regroupe les différentes options de contrôle de la charge utile, décrites ci-après, ainsi que les tâches de capture et de transmission des informations.

Nous considérons quatre modes de contrôle de la charge utile (figures 7.6b et 7.6c) :

Fix-Forward : la charge utile vise un point fixe devant l'avion (angle de 0° par rapport au cap de l'appareil) et ne peut être contrôlée directement. Ce mode est par exemple utilisé pour les phases d'atterrissage/décollage ou pour permettre à l'opérateur de réinitialiser l'orientation et l'inclinaison de la caméra lorsqu'il « a perdu le fil » après des phases de mouvements intempestifs.

Manuel : l'opérateur peut modifier l'orientation et l'inclinaison de la charge utile, ainsi que le niveau de zoom, afin de balayer la zone concernée. Dans les systèmes actuels, ce contrôle est effectué à l'aide d'un joystick.

Geotracking : la charge utile est verrouillée sur une position fixe, définie par des coordonnées spatiales (en trois dimensions).

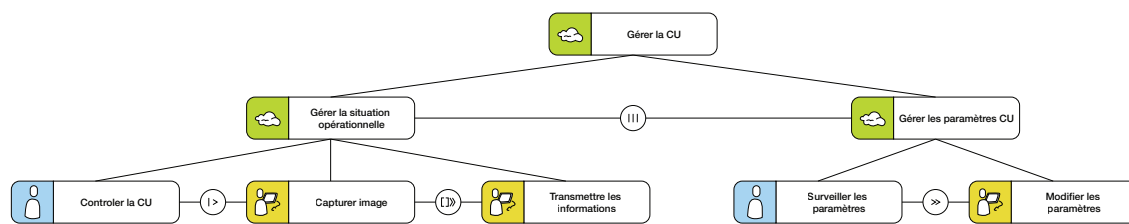
Suivi de piste : la charge utile est verrouillée sur une piste mobile, un algorithme spécifique est alors enclenché afin d'asservir l'orientation et l'inclinaison de la charge utile à la position de cet objet.

Pour ces quatre modes, l'opérateur doit sélectionner une optique parmi les deux installées dans la « boule » :

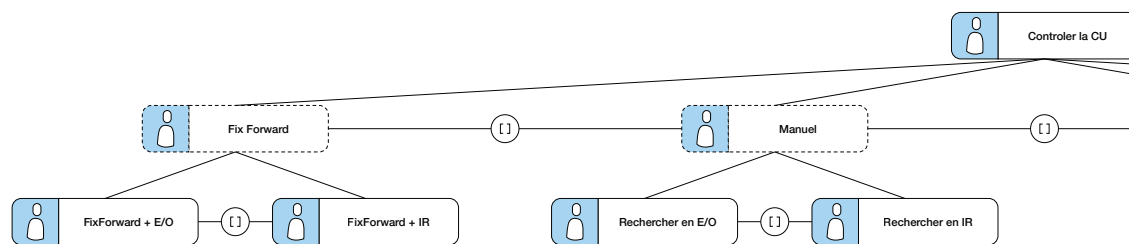
- L'**Electro/Optique**, dont l'image est en couleur, est généralement utilisée de jour ;
- L'**Infrarouge**, détectant la chaleur des objets, peut être utilisée aussi bien de jour que de nuit.

7.4.2 Modèle de Situation Awareness

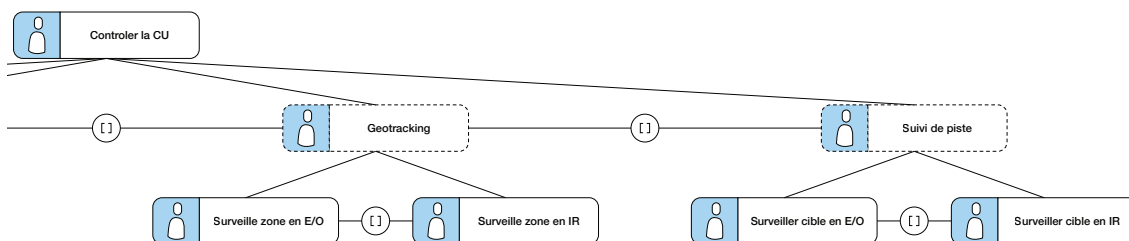
Le modèle de *Situation Awareness* de ce cas d'application est représenté par un réseau propositionnel composé de 117 nœuds et de 171 arêtes. Afin d'en simplifier la lecture, nous l'avons agencé en six groupes thématiques à l'aide de l'application *yEd Graph Editor*. Nous les décrivons dans ce paragraphe. La figure 7.7 représente



(a) les sous-tâches directes de la tâche « Gérer la Charge Utile ».



(b) les sous-tâche de la tâche « Contrôler la Charge Utile » (partie gauche).



(c) les sous-tâche de la tâche « Contrôler la Charge Utile » (partie droite).

FIGURE 7.6 : la tâche « Gérer la Charge Utile » et ses sous-tâches.

l'agencement de ces différents groupes et permet au lecteur de se repérer dans la lecture de ce graphe.

7.4.2.1 Éléments relatifs au drone

Cet ensemble d'éléments (figure 7.8) rassemble les informations relatives au vol du drone telles que le mode de pilotage activé (que l'opérateur doit connaître afin d'éviter les erreurs du type « confusion de mode »), la vitesse, la trajectoire verticale (montée / descente), la position dans l'espace (relative au sol et absolue). Les aspects contextuels du vol sont également importants, notamment la trajectoire passée (qui permet à l'opérateur de savoir quelles zones ont été traitées) ou les portées de la charge utile pour les trois objectifs DRI (Détection / Reconnaissance / Identification). Les informations sur la position des parties du drone (ailes, train d'atterrissage) par rapport à la caméra sont également importantes car ces dernières peuvent venir masquer l'image, engendrant une interruption temporaire dans la mission : un opérateur sachant où est située l'aile par rapport à l'orientation de la charge utile peut anticiper lors d'un virage et éviter le passage de l'aile dans le champ de la caméra.

7.4.2.2 Éléments relatifs à la météorologie

Cet ensemble d'éléments permet de regrouper les différents aspects météorologiques de l'environnement (figure 7.9). Ces éléments ont un impact sur la perception d'informations d'autres groupes : le soleil peut produire des reflets masquant la caméra, le vent peut impacter le pilotage, etc. Ces impacts sont traduits par les arêtes sortant du groupe (figure 7.7).

7.4.2.3 Éléments relatifs à la charge utile

Les éléments concernant la charge utile (figure 7.10) décrivent le mode de contrôle courant de la caméra, le type d'optique utilisée et les différents paramètres nécessaires à l'utilisation optimale de la charge utile. Ce groupe d'éléments contient également les informations portant sur l'orientation de la charge utile, la zone visée au sol (« trace capteur ») ou l'historique des zones balayées. Les informations permettant de faire remonter les pannes de la charge utile sont également représentées, sans en détailler les différents aspects (que nous avons laissés de côté pour cette analyse).

7.4.2.4 Éléments relatifs à la transmission d'informations

Ce groupe rassemble les différents interlocuteurs et les éléments à envoyer à ces interlocuteurs (figure 7.11). Il peut s'agir de l'analyste image, situé dans la station sol aux côtés de l'opérateur, ou de la hiérarchie. Dans certains cas, l'opérateur peut transmettre des images ou directement le flux vidéo à un terminal vidéo distant, le

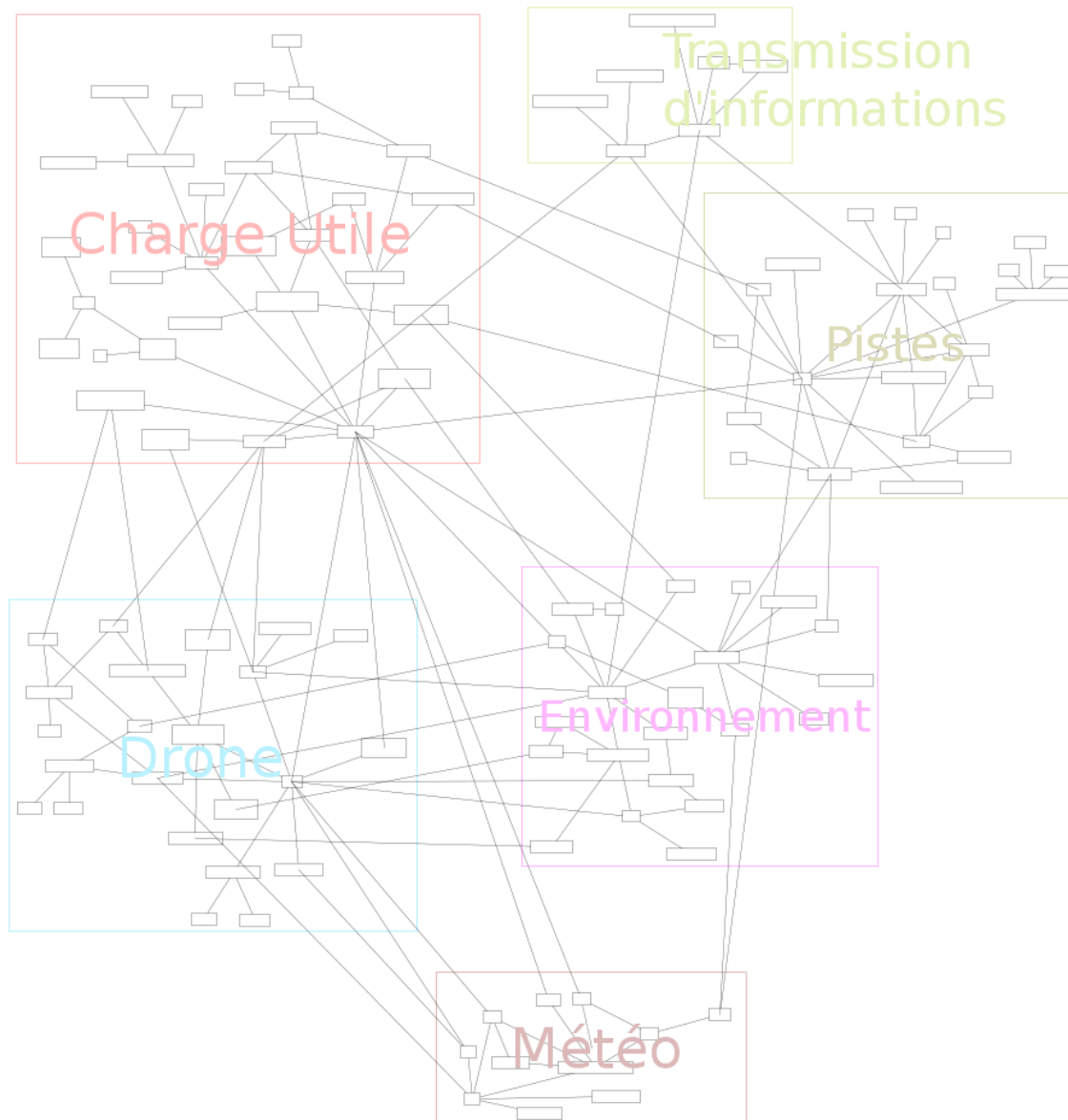


FIGURE 7.7 : les six groupes du graphe global de *Situation Awareness*.

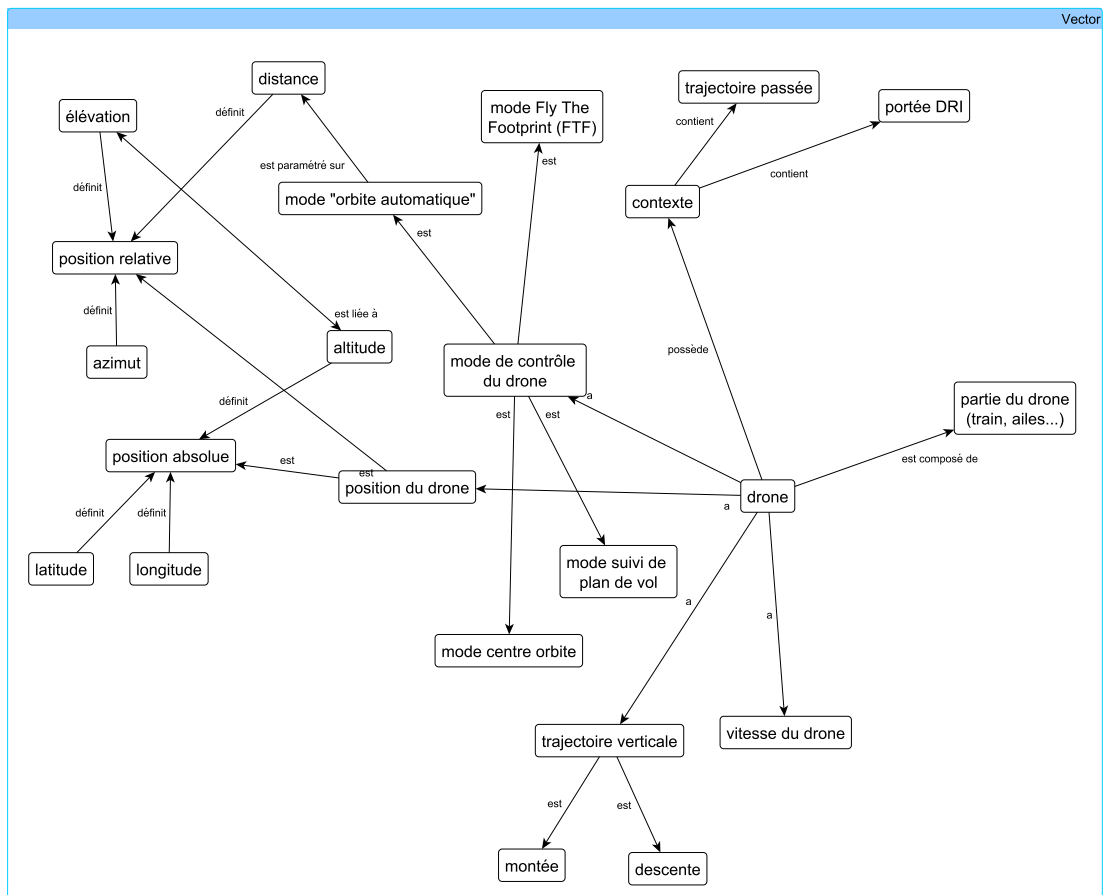


FIGURE 7.8 : les éléments du graphe global relatifs au drone.

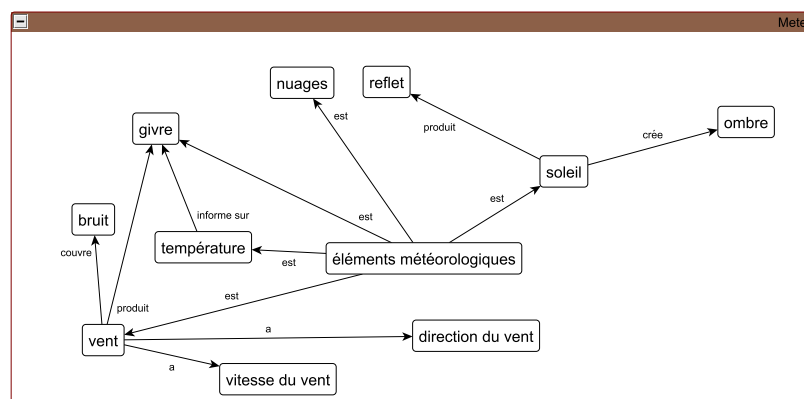


FIGURE 7.9 : les éléments du graphe global relatifs à la météorologie.

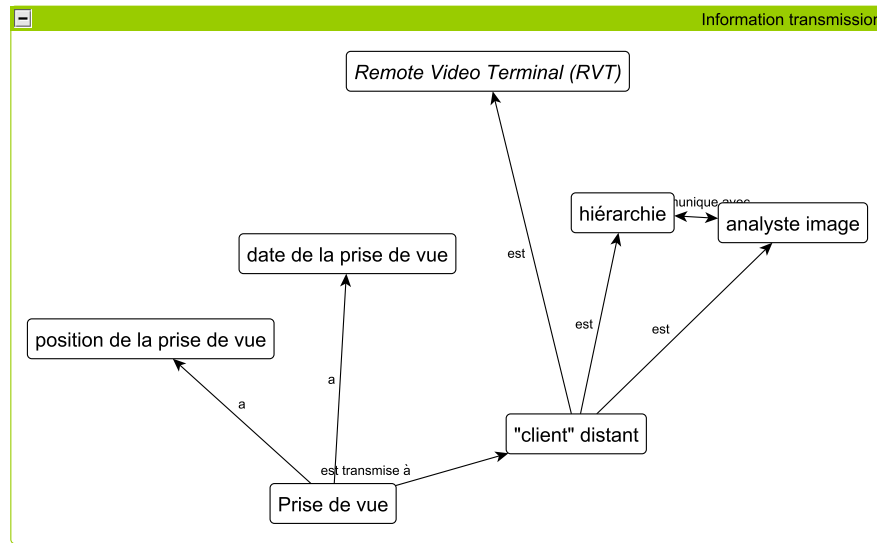


FIGURE 7.11 : les éléments du graphe global relatifs à transmission des informations

RVT (*Remote Video Terminal*). Cet élément est actuellement utilisé dans l'Armée de Terre avec le système SDTI de Sagem.

7.4.2.5 Éléments relatifs aux pistes

Ce groupe rassemble toutes les informations utiles à l'opérateur pour la détection, la reconnaissance et l'identification des pistes sous le drone (figure 7.12). Le terme « piste » décrit tous les êtres vivants et les véhicules pouvant être détectés par la charge utile, c'est un terme traduit du mot anglais « *track* » et couramment utilisé dans le domaine du renseignement militaire. Les informations de comportement et de type définissent le statut (ami, ennemi, neutre) de la piste. La dynamique de la piste est également importante lorsqu'elle est surveillée. Les éléments saillants à détecter dépendent également de l'optique utilisée (couleur en électro/optique, chaleur en infrarouge).

7.4.2.6 Éléments relatifs à l'environnement

Ce groupe (figure 7.13) rassemble à la fois les éléments de situation relatifs aux points de repère de l'environnement (bâtiments, routes, plans d'eau...) et les informations nécessaires à la mission et à la description des objectifs (points d'intérêts, zones restreintes de vol ou à surveiller...). Les informations de relief font également parties de ce groupe, elles restreignent notamment le vol du drone ainsi que le contrôle de la charge utile.



7



7

7.5 DÉFINITION ET VALIDATION OPÉRATIONNELLE DES CONTEXTES

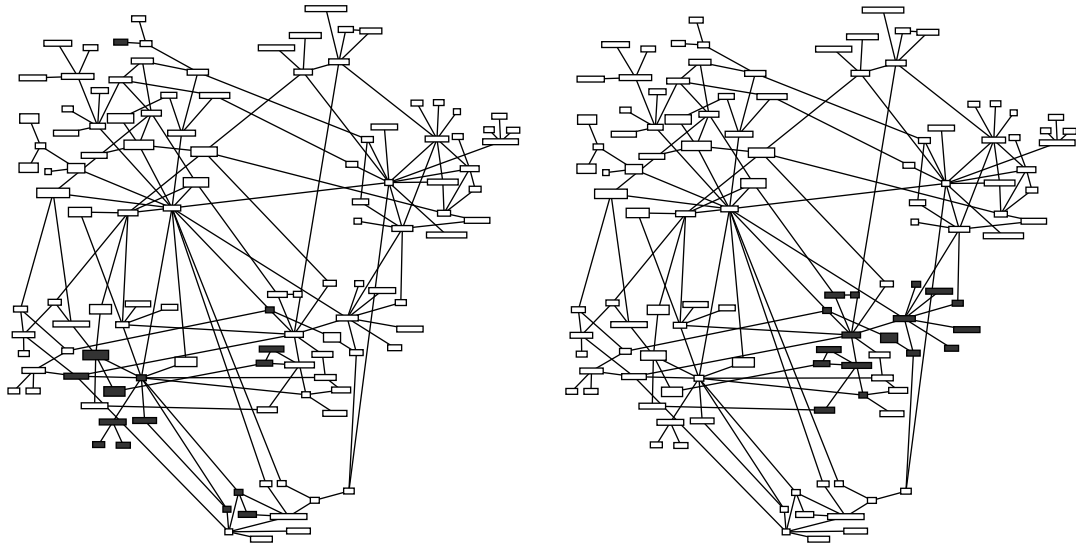
Conformément au déroulement de la méthodologie SAMANTA, l'étape suivante consiste à reprendre les données collectées et à définir les contextes associés à chacune des tâches de l'arbre. Par exemple, la figure 7.14 présente un aperçu des contextes définis pour les tâches « Suivi de plan de vol » (figure 7.14a), « S'orienter dans l'environnement » (figure 7.14b) et « Rechercher en E/O » (figure 7.14c). L'ensemble des contextes définis pour ce cas d'étude, participant à l'analyse des transitions, est présenté en détail dans l'annexe E.

Afin de compléter et valider les contextes définis lors de cette étape, nous avons réalisé des entretiens avec deux experts spécialisés dans la gestion de la charge utile. Ceci nous a permis de compléter les données collectées à travers le PEA FH/PA, où le focus était plutôt sur la gestion et le pilotage du vecteur.

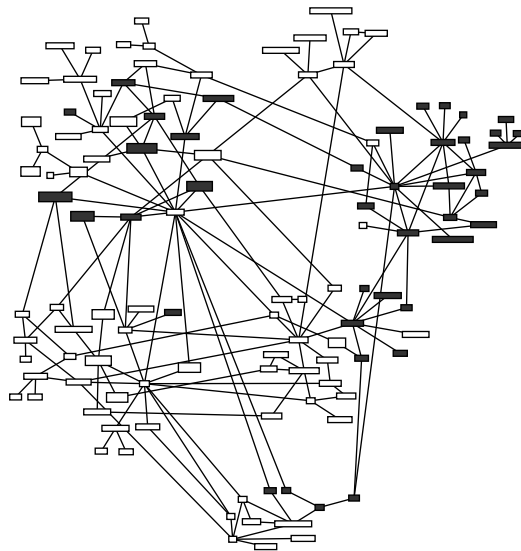
- Le 5 avril 2011, nous avons rencontré un **ingénieur d'essai de Cassidian AS**, spécialisé sur la gestion de la charge utile du SIDM (système MALE Harfang). Cet ingénieur a participé à des missions de support technique à l'Armée de l'Air lors de son déploiement en Afghanistan. Il fut également ingénieur d'étude chez Sagem DS sur les aspects « charge utile » du drone Sperwer, pour lequel il a été formateur auprès des opérateurs canadiens. Cet entretien s'est déroulé à Élancourt (78), dans les locaux de Cassidian AS.
- Le 7 avril 2011, nous avons rencontré un **commandant de l'escadron drone de Mont-de-Marsan** (Armée de l'Air). Cet escadron est déployé sur la Base Aérienne 709 de Cognac. Ancien navigateur de combat sur Mirage 2000 (modèles N et D) sur lequel il gérait notamment le capteur, il est maintenant opérateur pilote / charge utile sur drone Harfang. Il a participé à plusieurs opérations en Afghanistan dans le cadre du déploiement des forces françaises dans ce pays. Cet entretien s'est déroulé à Cognac (16), dans les locaux de l'Armée de l'Air, sur la base aérienne 709.

Les entretiens, d'une durée d'une heure et quinze minutes chacun environ, ont été menés sous une forme semi-guidée : après une présentation de la part des experts, une discussion était engagée sur la gestion du drone et plus particulièrement de la charge utile lors des différentes phases de mission. Dans un second temps, les modèles construits en amont étaient présentés pour nourrir la discussion, en demandant notamment aux experts quels étaient les éléments de situation exigés pour la réalisation des tâches identifiées.

Ces entretiens ont permis de compléter principalement le réseau propositionnel de situation, en y ajoutant les informations météorologiques, une partie des informations sur les pistes (*tracks*) importantes dans les phases de renseignement, et quelques informations sur l'environnement. Les contextes associés à chaque tâche de l'arbre ont également été complétés par ces entretiens.



(a) associé à la tâche « Suivi de plan de vol ». (b) associé à la tâche « S'orienter dans l'environnement ».



(c) associé à la tâche « Recherche en E/O ».

FIGURE 7.14 : aperçu des contextes définis pour quelques tâches du scénario.

7.6 ANALYSE DE TRANSITIONS : CHOIX DU SCÉNARIO ET DES TRANSITIONS

Le scénario type considéré pour notre cas d'application se focalise sur la phase d'exploration de zone et plus particulièrement sur deux transitions : un changement d'optique et un changement de mode de contrôle de la charge utile.

7.6.1 Description du scénario

Le drone doit effectuer une mission de renseignement sur une zone située autour d'un village suspecté d'abriter des soldats d'une organisation terroriste et de dissimuler une cache d'armes pour ce mouvement. Le décollage et le transit du drone de la base jusqu'à la zone se déroule sans encombre, en mode automatique, sur un plan de vol prédéfini lors de la préparation de mission.

1. A l'arrivée sur zone, l'opérateur passe en **mode de pilotage « orbite automatique »**, asservi sur la charge utile qu'il **contrôle en mode manuel** et avec l'optique « jour » (**Electro/Optique**). Il détecte un regroupement de personnes au sol.
2. Le crépuscule dégradant la visibilité, l'opérateur passe en mode **Infrarouge** afin de confirmer les informations et permettre à l'analyste image de transmettre des données plus précises à la chaîne de commandement. Un véhicule de type « pick-up », à l'arrière duquel sont assises plusieurs personnes portant des armes est détecté. Celui-ci se dirige vers la sortie du village.
3. Sur ordre de la chaîne de commandement, l'opérateur décide de suivre ce véhicule et enclenche le **mode « suivi de piste » de la charge utile**, verrouillé sur le véhicule léger. Ce mode est combiné à un **pilotage en mode « Fly The Footprint »** (le drone suit automatiquement la direction de la trace capteur au sol).
4. Après quelques minutes de surveillance, le véhicule change de direction à un carrefour. Dans le virage effectué par le vecteur afin de suivre le mouvement du véhicule, l'aile du drone vient masquer la charge utile et l'algorithme de suivi de piste « décroche », **repassant en mode manuel**. L'opérateur doit alors compenser à l'aide du joystick afin de conserver une visée sur la cible. Il décide de conserver la visée manuelle pour la suite de la mission, ne faisant plus confiance à l'algorithme.
5. Le pick-up se dirige vers une zone montagneuse où la couche nuageuse est trop dense pour permettre à l'opérateur de poursuivre cette mission de surveillance. L'information est transmise à la chaîne de commandement, qui décide d'envoyer une patrouille au sol. Le drone retourne vers la zone de recherche initiale, en mode suivi de plan de vol, où il reprendra sa mission de surveillance du village.

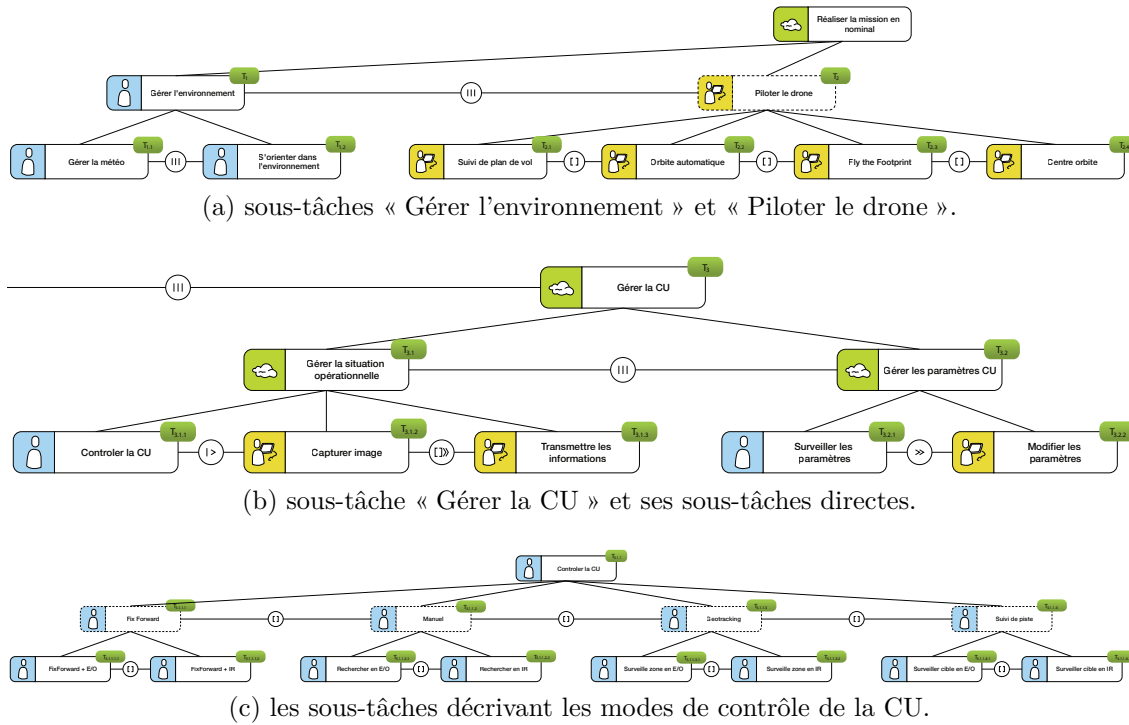


FIGURE 7.15 : le modèle de tâches et les étiquettes formalisées associées.

7.6.2 Formalisation du scénario

Dans ce paragraphe, nous formalisons l'évolution des tâches, décrite dans le scénario précédant, en utilisant la notation proposée dans le paragraphe 5.2.2.6, où chaque tâche est notée sous la forme T_i .

A cette fin, l'arbre de la figure 7.3 est étiqueté avec ce formalisme, et présenté dans la figure 7.15.

La tâche « Gérer l'environnement » (T_1) est active de manière continue pendant tout le scénario.

Lorsque l'opérateur utilise la charge utile, la tâche « Gérer les paramètres CU » ($T_{3.2}$) vient également s'ajouter à l'une des sous-tâches de contrôle de la CU. La trame principale du scénario présenté peut être divisée en plusieurs états, auxquels sont associés des ensembles de tâches.

État initial : à l'arrivée sur la zone, le drone est piloté en mode « suivi de plan de vol » ($T_{2.1}$) et nous considérons que l'opérateur ne se focalise pas sur la caméra, qui est en mode « *Fix Forward E/O* » ($T_{3.1.1.1.1}$). Ainsi, l'état initial correspond à l'ensemble de tâches $T_1, T_{2.1}, T_{3.1.1.1.1}$.

Surveillance du village E/O : le premier changement d'état se traduit par l'enclenchement du mode de pilotage en « orbite automatique » ($T_{2.2}$) et l'activation du contrôle manuel, en Electro/Optique, de la charge utile ($T_{3.1.1.2.1}$). Le nouvel état correspond à l'ensemble des tâches $T_1, T_{2.2}, T_{3.1.1.2.1}$.

7.6. ANALYSE DE TRANSITIONS : CHOIX DU SCÉNARIO ET DES TRANSITIONS

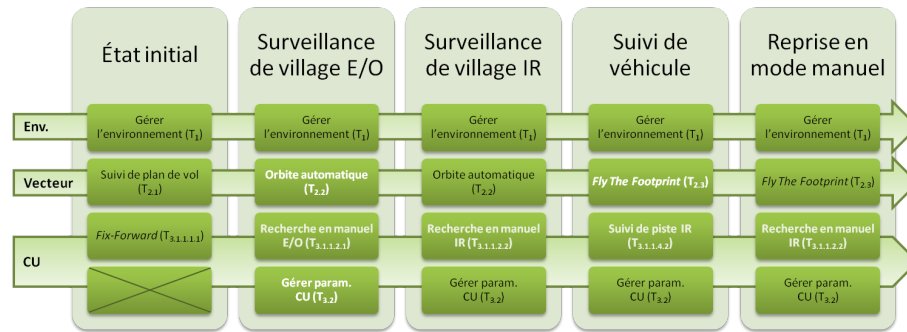


FIGURE 7.16 : représentation des états du scénario et des tâches associées.

Surveillance du village IR : la baisse de visibilité entraîne le changement d'optique, de l'Electro/Optique vers l'Infrarouge, tout en conservant le mode de contrôle manuel ($T_{3.1.1.2.2}$). Les autres tâches restent inchangées, le nouvel état est décrit par l'ensemble des tâches $T_1, T_{2.2}, T_{3.1.1.2.2}$.

Suivi du véhicule : le départ du véhicule entraîne un passage en mode « suivi de piste », toujours à l'aide de l'optique infrarouge ($T_{3.1.1.4.2}$). Ce mode est associé à un pilotage asservi sur la charge utile en mode « *Fly The Footprint (FTF)* » ($T_{2.3}$). Le nouvel état est décrit par l'ensemble $T_1, T_{2.3}, T_{3.1.1.4.2}$.

Reprise en mode manuel : le masquage de la vue par l'aile de l'appareil déverrouille le mode « suivi de piste » et entraîne le passage en mode manuel ($T_{3.1.1.2.2}$). L'opérateur continue à suivre la cible en pilotant le drone par un asservissement sur la trace senseur ($T_{2.3}$). Le nouvel état est décrit par l'ensemble $T_1, T_{2.3}, T_{3.1.1.2.2}$.

L'abandon de la mission de suivi de cible se traduit par le retour en phase de « transit » : l'opérateur crée généralement un plan de vol réduit retournant vers le village, où il reprendra sa tâche dans un mode de surveillance infrarouge.

A cet enchaînement séquentiel viennent s'ajouter les interruptions entraînées par la capture et la transmission des prises de vue au cours de la mission. Ces interruptions peuvent intervenir à plusieurs reprises au cours des phases de surveillance. La figure 7.16 présente la formalisation de la trame scénaristique présentée.

Nous remarquons que le passage de l'état initial vers la surveillance du village en E/O et celui de la surveillance en IR vers le suivi de véhicule impliquent plusieurs changements de tâches (apparition de la tâche « Gérer paramètres CU », changement des modes de pilotage et de contrôle de la CU). Ces changements de tâche peuvent être décrits par une suite de transitions atomiques où la priorité est donnée au changement de tâche concernant le mode de pilotage (afin d'assurer l'intégrité de l'appareil), suivi du changement de mode de contrôle de la charge utile. Les états intermédiaires ainsi créés n'existent en réalité que pendant un court laps de temps (figure 7.17).

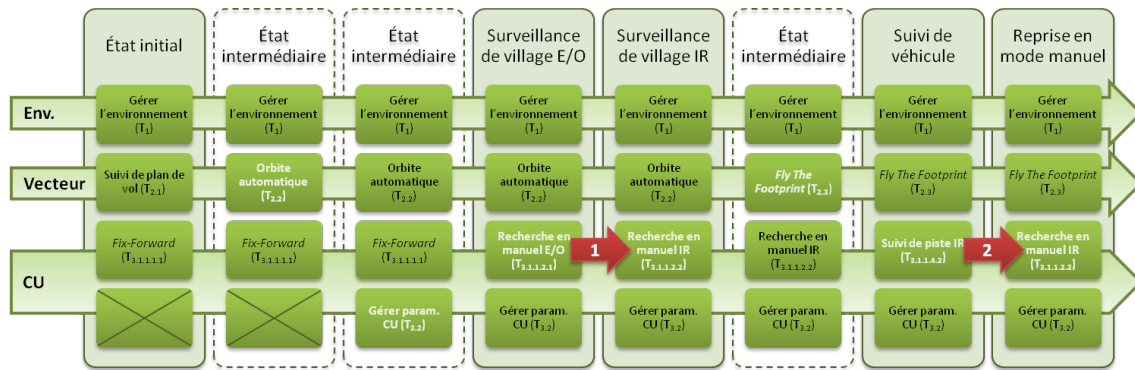


FIGURE 7.17 : représentation des états du scénario et des transitions « atomiques » entre ces états.

Nous proposons de nous intéresser à deux transitions spécifiques de ce scénario :

- le changement d'optique effectué lors de la surveillance du village, cette transition est déclenchée par l'opérateur (transition n°1 sur la figure 7.17) ;
- le changement de mode de contrôle de la charge utile, déclenché par le masquage de la charge utile par l'aile et le déverrouillage de l'algorithme de suivi de piste pour repasser en mode manuel (transition n°2 sur la figure 7.17).

Nous avons choisi ces transitions pour deux raisons :

- Elles ont été mises en avant par les opérateurs lors des discussions comme ayant un impact important sur leur *Situation Awareness*. En effet, il s'agit de transitions moins fréquemment rencontrées, et, quand elles le sont, qui apparaissent dans des phases de mission où la charge de travail de l'opérateur est généralement plus importante.
- Ces deux transitions ont un rapport direct avec les tâches de manipulation de la charge utile, sur lesquelles nous nous sommes focalisés lors des entretiens, les experts rencontrés étant spécialisés sur la partie charge utile des systèmes de drone plutôt que sur la partie pilotage.

Ainsi, dans la suite de ce chapitre, nous présentons l'analyse de ces deux transitions, où nous proposons des recommandations de solutions IHM ou technologiques afin de répondre aux problématiques rencontrées.

7.6.3 Analyse de transition #1 : changement d'optique en contrôle manuel de la charge utile

La transition analysée dans ce paragraphe consiste à changer l'optique activée dans la charge utile et dont le flux vidéo est transmis à l'opérateur. En effet, nous considérons que pour ce système, un seul flux vidéo peut être envoyé à l'opérateur à la fois.

La configuration initiale d'utilisation est l'Electro/Optique, qui fonctionne dans

le spectre de lumière visible. Le flux vidéo reçu par l'opérateur est en couleurs et permet d'observer la situation visée par la charge utile comme avec une caméra « classique ».

La nouvelle configuration est l'Infrarouge, détectant les émissions dans ce domaine spectral. Cette configuration permet ainsi de mettre en avant les corps et objets émettant de la chaleur. Le flux vidéo renvoyé présente une image en niveaux de gris, les points les plus chauds pouvant être affichés :

- en blanc (polarité *white hot*), avec un dégradé en niveaux de gris vers le noir lorsque la chaleur émise diminue. Ce mode facilite la détection des pistes qui « rayonnent » sur l'image.
- en noir (polarité *black hot*), avec un dégradé en niveaux de gris vers le blanc lorsque la chaleur émise diminue. Ceci permet d'avoir plus de détails en niveaux de gris sur l'image, et facilite l'identification des cibles (le blanc de l'image ne vient pas « éblouir » l'opérateur lors de l'identification).

L'utilisation du capteur Infrarouge en journée permet par exemple de faciliter la détection de pistes situées sous la végétation feuillue ou en train d'utiliser des armes à feu. Il sera toujours utilisé la nuit.

Nous notons que les deux tâches impliquées dans cette transition ont la même tâche parente. En effet, pour ces deux tâches, l'opérateur contrôle sa charge utile en mode manuel, et doit donc contrôler directement l'orientation et l'inclinaison de la charge utile (ce qui se fait généralement à l'aide d'un joystick). Ceci implique que dans ce cas, l'interaction est la même pour les deux tâches, seul le flux d'information revenant vers l'opérateur est différent, et les éléments perçus par l'opérateur n'ont plus la même importance en fonction du capteur sélectionné.

7.6.3.1 Statistiques générales

Il y a 13 éléments de situation propres au capteur Electro/Optique, 8 éléments propres au capteur Infrarouge et 28 éléments partagés par les deux capteurs. Ainsi, si l'on considère que les deux tâches ont la même importance dans la transition ($\alpha = \beta = 1$), nous obtenons un score de Tversky de 57 %.

Dans la taxinomie des transitions (section 4.1), la transition considérée est une transition simple avec intersection de contextes (figure 7.18). Nous revenons sur la nature des éléments partagés dans le paragraphe 7.6.3.2.

Éléments propres à la tâche « Rechercher en E/O »

Les éléments de situation attachés à la tâche « Recherche en E/O » uniquement sont les éléments : *bâtiment*, *point de repère*, *forme géométrique*, *champ*, *route*, *carrefour*, *ombre*, *soleil*, *reflet*, *Electro/Optique (E/O)*, **netteté**, *niveau de zoom*, *couleur*. Nous constatons qu'il s'agit d'éléments relatifs à l'observation dans le spectre visible.

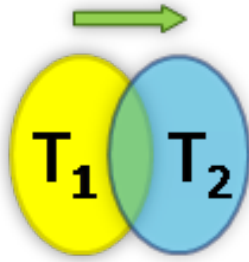


FIGURE 7.18 : la première transition considérée est une transition simple, avec intersection de contextes.

Les éléments relatifs aux *points de repères* constituent pour les opérateurs, lors de la recherche en mode manuel E/O, des endroits possibles de détection de l'activité : des travailleurs dans des champs, des personnes sur les routes ou aux carrefours, etc. Tous ces éléments sont reliés aux éléments de situation relatifs aux pistes et dirigent la recherche avec le capteur E/O mais ne sont plus nécessaires avec le capteur IR, car plus difficiles à détecter avec ce capteur.

Néanmoins, ces éléments appartiennent également au contexte de la tâche « S'orienter dans l'environnement » s'exécutant en parallèle. A ce titre, même s'ils ne sont plus pertinents pour la tâche principale courante, il peut être intéressant de permettre à l'opérateur de conserver une conscience de ces éléments, par l'intermédiaire de la cartographie et d'un modèle en trois dimensions du terrain.

De la même façon, certains éléments relatifs à la *météo* sont propres à l'utilisation du capteur E/O car ils influencent la visibilité de l'opérateur, soit en créant des reflets (soleil sur une étendue d'eau, par exemple) ou en masquant les potentielles pistes au sol sous l'ombre d'un bâtiment, ombre qui n'aura aucune influence avec le capteur IR.

Enfin, quelques éléments sont propres fonctionnement avec l'Electro/Optique : le paramètre de netteté (*sharpness*) est un paramètre très souvent modifié lors de la surveillance en E/O. Il est d'ailleurs saillant pour ce mode, mais il est moins important en IR, où l'opérateur jouera plutôt sur le contraste de l'image. De la même façon, la couleur des pistes n'est plus un élément pertinent au profit de leur chaleur. Le paramètre saillant de *niveau de zoom* exprime le fonctionnement continu du zoom de la caméra E/O. Cet élément n'existe plus en IR, où la caméra fonctionne avec un zoom par palier.

Nouveaux éléments, propres à la tâche « Rechercher en IR »

Les éléments de situation associés à la tâche « Rechercher en IR » uniquement sont les éléments : *polarité*, *Infrarouge (IR)*, *palier de zoom*, *white hot*, ***black hot***, *niveau de contraste*, *paramètre de contraste*, ***chaleur***. Nous retrouvons uniquement

7.6. ANALYSE DE TRANSITIONS : CHOIX DU SCÉNARIO ET DES TRANSITIONS



FIGURE 7.19 : illustration de la différence d'image pour une même visée avant et après le changement d'optique.

des éléments associés au paramétrage de la charge utile pour ce capteur.

Concernant les paramètres de polarité, notons que l'élément *black hot* est saillant. En effet, il s'agit de la configuration préférée des opérateurs interviewés, car elle permet d'obtenir un niveau de détails plus élevé sur les pistes détectées, et permet leur identification. Cette préférence peut-être traduite simplement en définissant cette configuration comme configuration « par défaut » lors du changement d'optique. Notons que ces préférences, issus de deux entretiens, ne constituent peut-être pas une généralité et des tests de l'interface devront permettre de confirmer ou infirmer ce point.

Parmi les nouveaux éléments de situation, l'élément *palier de zoom*, qui exprime le fait que le capteur IR fonctionne avec un zoom par palier, est un élément frontière avec l'élément *niveau de zoom*, exprimant le fonctionnement en zoom continu du capteur E/O. Lors d'un changement d'optique, le passage d'un type de zoom à l'autre doit se faire de la manière la plus fluide possible. En particulier, si le niveau de zoom continu en E/O ne correspond à aucun des paliers du zoom en IR, le changement peut induire un « saut » de l'image, vers le palier supérieur ou inférieur, nécessitant à l'opérateur de se recalibrer mentalement sur la situation. La figure 7.19 présente la différence entre ces deux images, extraite d'une vidéo commerciale² du constructeur de charge utile Wescam³.

Plusieurs solutions permettent de pallier ce changement brusque de niveau de zoom :

- Technologiquement parlant, une optique infrarouge avec un zoom continu peut être embarquée. Dans ce cas, les niveaux de zoom des deux optiques devront être les mêmes à tout instant et donc, la commande de zoom transmise à

2. Source : <http://www.youtube.com/watch?v=XXuWnjFzIz0>, images extraites à 4:04 et 4:05.

3. http://www.wescam.com/products/products_services_1g.asp



FIGURE 7.20 : proposition d'aide à la transition pour pallier les différences de zoom entre les capteurs E/O et IR.

l'optique courante devra être également appliquée à la deuxième optique. Ces deux optiques sont installées côte à côte dans la « boule » (il s'agit de deux capteurs disjoints). Ainsi, lors d'un changement d'optique, la deuxième optique aura le même niveau de zoom que celle utilisée auparavant.

- Dans le cas où le zoom par palier est conservé pour l'optique IR, nous envisageons deux solutions.
 - Empêcher le changement d'optique lorsque le niveau de zoom en E/O ne correspond pas à un palier du capteur IR. Dans ce cas, il faut que cet empêchement soit explicite au niveau de l'interface et que l'opérateur en comprenne la cause et soit accompagné dans la résolution de cette incompatibilité.
 - Lorsque les deux niveaux de zoom sont différents, une aide à la transition peut être affichée sur l'interface. Il peut s'agir d'une animation entre les deux flux vidéos, permettant à l'opérateur de faire le lien entre les éléments de leurs images, ou encore l'affichage temporaire d'un cadre sur la vidéo initiale correspondant à la zone qui sera affichée à l'écran après la transition (figure 7.20).

7.6.3.2 Éléments partagés et sous-graphes communs

Éléments partagés

Parmi les 28 éléments de situation partagés entre les deux tâches de la transition, 3 d'entre eux sont des éléments indépendants du graphe, c'est-à-dire qu'ils ne sont reliés à aucun autre élément partagé. Il s'agit des éléments *nuages*, *type de charge utile* et *portée DRI*.

L'élément *type de charge utile* contient l'information de l'optique active pour le flux vidéo visualisé par l'opérateur. Cette information transparait directement sur l'interface où elle a un impact sur l'aspect de l'image (couleur ou niveaux de gris).

L'information concernant les *nuages* est importante car elle peut venir gêner la vue de l'opérateur sur la scène observée. Il s'agit d'une information déjà présente dans le contexte associé à la gestion de la météo, que l'opérateur doit prendre en

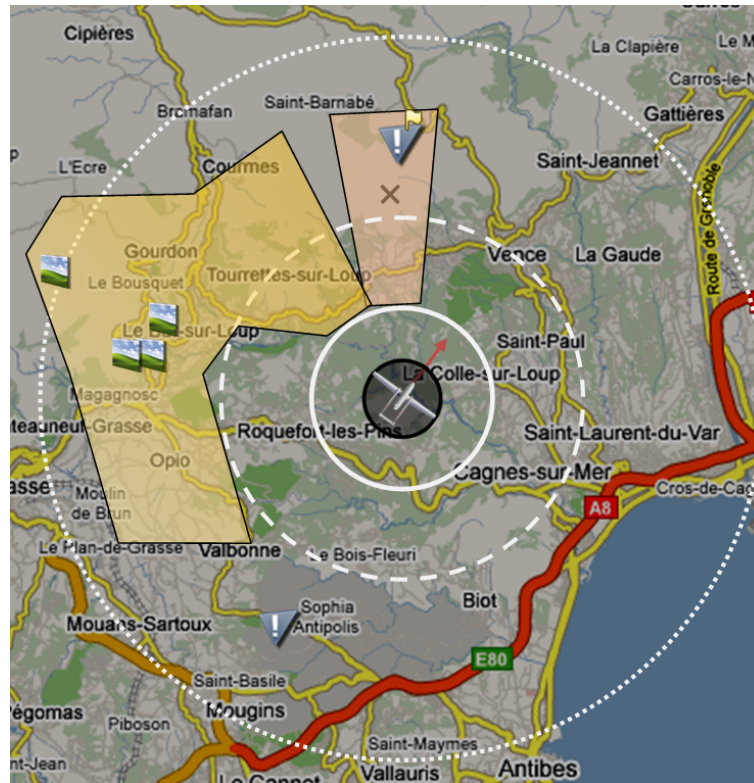


FIGURE 7.21 : affichage des zones DRI (cercles concentriques blancs) et de l'historique de la trace capteur (en dégradé jaune) sur une maquette de la vue cartographique.

compte constamment, par l'intermédiaire de cartes météorologiques, pouvant être mises à disposition sous la forme de calques sur la vue cartographique. Ces informations lui permettront de décider de l'approche à adopter pour l'observation d'une zone et éventuellement de l'altitude du drone.

L'information de *portée DRI* est une information de distance relative par rapport au drone. Elle permet de définir les zones autour du drone permettant de réaliser une Détection de piste, une Reconnaissance (statut de la piste) et une Identification (qui est la piste?). Cette information est définie par 3 cercles concentriques, centrés sur le drone, représentant des distances théoriques pour chaque type de portée. Le changement d'optique n'a pas d'effet sur cette donnée, qui pourra être affichée sur la cartographie par exemple (figure 7.21).

Plus grands sous-graphes communs

Les autres éléments de situation partagés par les deux tâches forment deux sous-graphes communs, un rassemblant des éléments portant sur le contrôle de la charge utile en manuel (figure 7.22), et le deuxième décrivant les éléments relatifs aux pistes

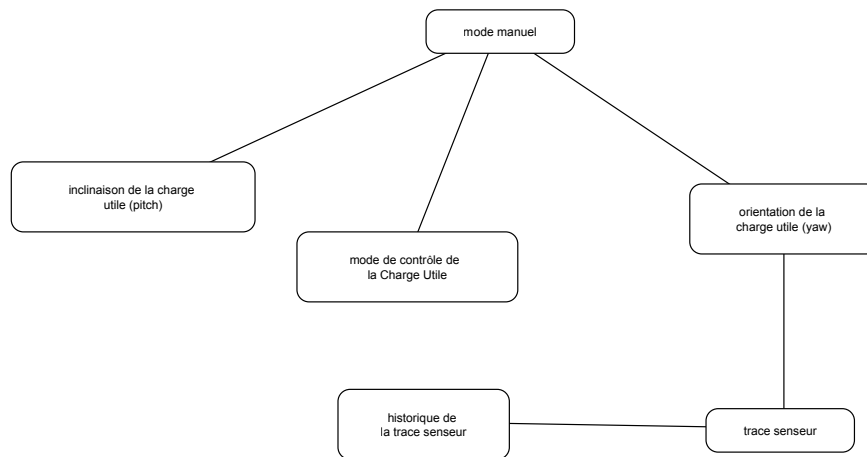


FIGURE 7.22 : sous-graphe associé aux éléments de contrôle manuel de la charge utile.

observées (figure 7.23).

Les deux tâches étudiées faisant partie du même mode de contrôle de la charge utile (mode manuel), il est cohérent de retrouver un ensemble d'éléments partagés relatifs à ce mode. L'information de mode courant (éléments *mode de contrôle de la Charge Utile* et *mode manuel*), peut être importante afin d'éviter les erreurs dues à la confusion de mode (ce qui est peu probable dans le cas d'une transition non interrompue et déclenchée par l'opérateur). Cet élément est saillant selon la définition de Stanton.

En outre, les informations relatives à *l'orientation* (angle sur le plan parallèle au sol) et *l'inclinaison de la charge utile* (angle autour d'un axe perpendiculaire à la direction du vecteur) sont importantes dans les deux cas de figure.

En effet, les deux optiques n'étant pas positionnées exactement au même endroit dans la « boule », il existe un décalage de quelques degrés entre les deux visées. Sur les systèmes actuels étudiés, ce décalage n'est pas compensé et produit le même effet de « saut d'image » que la différence de fonctionnement de zoom (voir paragraphe précédent). Ainsi, afin d'assurer une continuité sur ces informations lors de la transition, il faut mettre en œuvre le même type de solution que dans le cas de la différence de zoom.

- Mettre en place un asservissement technologique des optiques, afin, lors d'un changement d'optique, de compenser les quelques degrés de décalage et de conserver la même visée. L'orientation et l'inclinaison de la boule doivent alors être modifiés automatiquement de manière transparente pour l'utilisateur. Néanmoins, même s'il est mis en place, ce type de transition « mécanique » ne se fait pas instantanément et l'opérateur verra un léger décalage

sur la vidéo lors du changement d'image.

- Accompagner la transition sur l'interface en calant les deux images sur des éléments partagés de la scène, ou en accompagnant la transition par une animation entre les deux images décalées. Ceci est plus aisé à mettre en place lorsque le niveau de zoom est faible : s'il est trop élevé, le faible angle « plan » entre les deux optiques peut correspondre à un angle « solide » important (angle projeté sur la sphère terrestre).

Enfin, les éléments relatifs à la trace capteur (la projection de la vue caméra sur le sol) sont attachés aux contextes des deux tâches de la transition. Nous notons que l'élément *trace capteur* est saillant au sens de Stanton (il a plus de cinq éléments liés à lui dans le graphe global). L'information de trace au sol est importante car elle permet à l'opérateur de savoir où il vise, et où il a visé (zones observées) dans le cas de l'élément *historique de la trace capteur*. Dans le cas d'un changement d'optique, il peut être intéressant de faire la différence entre l'historique capteur construit en E/O et le nouvel historique en infrarouge (figure 7.21).

En effet, l'opérateur utilise cette information afin d'éviter de repasser deux fois au même endroit lors du balayage d'une zone. Mais dans le cas d'un changement d'optique, où l'infrarouge peut apporter des informations supplémentaires, il peut être intéressant de faire cette différence d'usage afin de permettre à l'opérateur de décider de repasser sur certaines zones déjà observées avec l'autre optique.

Le deuxième sous-graphe commun aux deux tâches de la transition concerne les éléments de situation décrivant les pistes au sol et permettant de définir leur statut (neutre, ami, ennemi).

L'élément de situation *position de la piste* est défini comme saillant par l'opérateur quelque soit l'optique utilisée. En effet, cette information est primordiale pour l'opérateur, qui doit se focaliser sur les différentes pistes au sol et maintenir la visée de la charge utile vers la piste « courante » tant que son statut n'a pas été défini. Lorsque des pistes ont été détectées et identifiées, leur position est généralement enregistrée par l'analyste image. Cette information est retransmise à l'opérateur charge utile par l'intermédiaire de l'écran cartographique. Ainsi, afin de réduire l'impact du changement d'optique et l'éventuelle perte de focus de l'opérateur sur une piste visée, il peut être intéressant lors d'une telle transition de remettre en avant les informations sur la position des dernières pistes identifiées. Un exemple serait de mettre en exergue ces points sur la cartographie (où est également affichée la trace au sol du capteur) ou de les afficher directement en réalité augmentée au sein de la vue caméra ou par des indications sur les bords de l'écran lorsqu'elles sont « hors champ » (figure 7.24).

Les éléments *statut de la piste*, *type de piste*, *véhicule* et *comportement* sont saillants selon la définition de Stanton (plus de cinq liens vers des éléments voisins

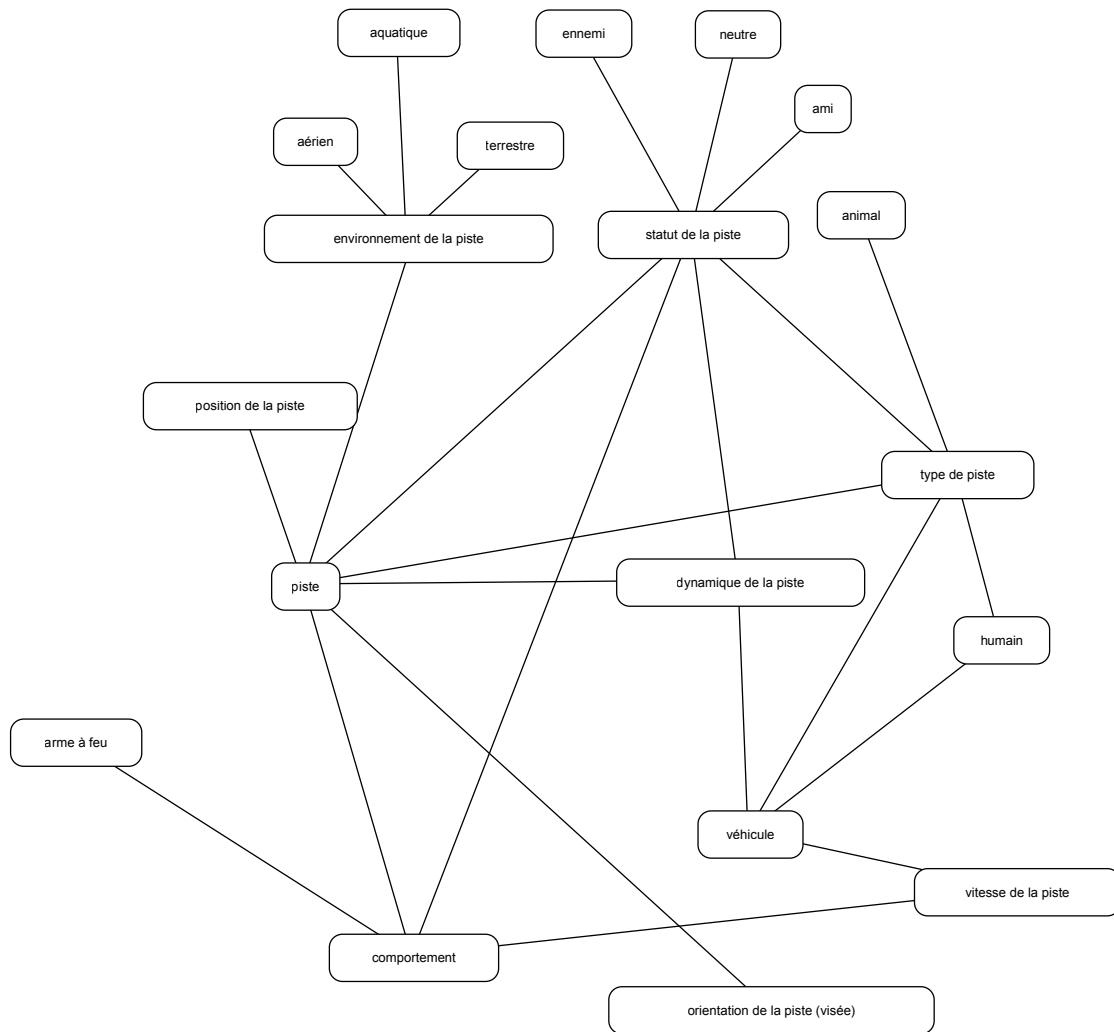


FIGURE 7.23 : sous-graphe associé aux différents aspects des pistes.



FIGURE 7.24 : indication sur la vue caméra de la direction des deux dernières pistes identifiées et « hors champ » (en haut à gauche).

du graphe global).

Le *statut de la piste* est une information définie par l'opérateur après étude d'un ensemble de paramètres, tels que son comportement, son type et son aspect dynamique. Le passage en infrarouge peut permettre à l'opérateur d'obtenir davantage de détails sur certaines pistes dont le statut n'aurait pas été défini par manque de détails ou de visibilité en E/O. En complément de l'information de position de ces pistes, la mise en exergue des pistes dont le statut est « inconnu » pourrait guider l'opérateur dans sa mission de recherche de cible.

Les informations de ce sous-graphe sont utiles quelque soit l'optique utilisée. Afin d'accompagner au mieux la transition et d'éviter des confusions de la part de l'opérateur, il faut s'assurer qu'elles soient affichées sur l'interface de la même manière (mêmes formes et couleurs de pictogrammes par exemple). En effet, comme nous l'avons vu dans le paragraphe 2.2.2.1, un stimulus univalent (affichage du même stimulus pour une même information à travers les tâches) réduit l'impact de la transition et les effets d'interférences (Allport *et al.*, 1994; Rogers et Monsell, 1995).

7.6.4 Analyse de transition #2 : changement de mode de contrôle de la charge utile

La deuxième transition analysée dans le cadre de cette application est un changement de mode de contrôle de la charge utile, passant de la tâche « Surveiller cible en IR », où la charge utile est verrouillée sur un véhicule mouvant, à la tâche « Rechercher en IR », où l'opérateur reprend la main sur le contrôle de la charge utile

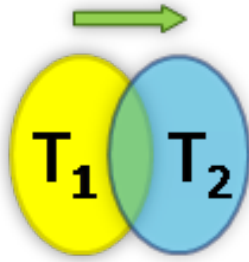


FIGURE 7.25 : la deuxième transition considérée est une transition simple, avec intersection de contextes.

en Infrarouge.

Cette transition peut être à l'initiative de l'opérateur, auquel cas il pourra s'y préparer. Mais dans certains cas, elle peut être déclenchée par le système. Indépendamment de l'origine de la transition, celle-ci induit un délai dans la reprise en mode manuel du suivi de l'objectif. Dans le cas où l'opérateur décide d'abandonner ce suivi, cela n'est pas important.

Mais dans le cas où le système déclenche la transition de manière imprévue, il peut s'écouler quelques secondes pendant lesquelles l'opérateur perd de vue le véhicule suivi. Il doit alors reconstruire une image mentale de la situation et notamment retrouver la position de ce véhicule. C'est ce dernier cas de figure que nous avons considéré dans cette analyse, en décrivant dans le scénario une transition déclenchée par le système suite à un « décrochage » de l'algorithme de suivi de piste, produit par le passage dans le champ de la caméra de l'aile du vecteur. Les recommandations issues de l'analyse porteront sur la façon dont l'interface peut aider l'opérateur à préparer la transition imprévue en conservant une SA sur les éléments importants, et comment l'impact éventuel sur sa SA peut être réduit lors du retour en contrôle manuel.

7.6.4.1 Statistiques générales

Il y a 11 éléments de situation propres au suivi de piste en infrarouge, 6 éléments propres au contrôle en mode manuel infrarouge et 30 éléments partagés par les deux modes. Ainsi, si l'on considère que les deux tâches ont la même importance dans la transition ($\alpha = \beta = 1$), nous obtenons un score de Tversky de 64 %.

Dans la taxinomie des transitions (section 4.1), la transition considérée est une transition simple avec intersection de contextes (figure 7.25).

Éléments propres à la tâche « Surveiller la cible en IR »

Les éléments de situation associés uniquement au contexte de la tâche « Surveiller en IR » sont au nombre de 11 : *relief*, *bâtiment*, *carrefour*, *point de repère*, *route*,

7.6. ANALYSE DE TRANSITIONS : CHOIX DU SCÉNARIO ET DES TRANSITIONS

hauteur du bâtiment, stabilisation de la caméra, mode verrouillage géolocalisé, distance, partie du drone (train, ailes...), position relative.

Certains de ces éléments se rapportent aux points de repère de l'environnement : *relief, bâtiment, carrefour, point de repère, route, hauteur du bâtiment*. L'élément *point de repère* est d'ailleurs saillant au sens de Stanton. Ces éléments sont pertinents lors de la réalisation de cette tâche car leur connaissance par l'opérateur lui permettent d'anticiper sur les mouvements de la cible suivie. Par exemple, sur une route droite, un véhicule changera rarement de direction tandis qu'à l'approche d'un carrefour, l'opérateur doit être plus attentif aux mouvements du véhicule. Autre exemple, lors d'une surveillance de convoi ami, les bâtiments peuvent abriter des ennemis et le passage du véhicule suivi à proximité de ces bâtiments peut présenter un danger.

Ainsi, dans le cadre d'une transition impromptue vers un mode de contrôle manuel, lorsque l'opérateur va tenter de revenir vers la piste initialement suivie, il aura tendance à chercher ces points de repère afin de s'orienter dans l'environnement, et permettre de retrouver la *position de la piste*, et des points de repères alentour à surveiller.

Afin de l'aider, des algorithmes de détection de contours de constructions comme ceux utilisés par l'IGN (Baillard et Maître, 1999) pourront par exemple être mis en œuvre. L'opérateur pourra également s'appuyer sur un modèle 3D de l'environnement, au sein duquel seraient affichées en temps réel la position du vecteur et la visée de la caméra. Ce type de vue 3D est déjà mis en œuvre sur le système Sperwer (version canadienne du drone SDTI de Sagem DS).

Notons que ces éléments permettent également d'appuyer la tâche « S'orienter dans l'environnement », tâche continue dans l'activité de l'opérateur.

En mode « suivi de piste », l'information *stabilisation de la caméra* est importante pour l'opérateur car une panne de ce système signifie qu'il va devoir reprendre le contrôle en mode manuel directement. Afin de lui permettre d'anticiper, et de comprendre la transition le cas échéant, l'état du système de stabilisation doit pouvoir être visible à tout moment sur l'interface de manière claire et non ambiguë. On pourra par exemple utiliser les couleurs définies dans la norme militaire MIL-STD-1472G, tableau n°15 (vert lorsque tout fonctionne, rouge lorsque le système est hors service) du DoD (2012).

Enfin, le dernier élément associé uniquement au suivi de piste et pouvant avoir un impact sur la transition est l'élément *partie du drone*. En effet, lors des entretiens, les opérateurs ont indiqué devoir évaluer en temps réel la position des différentes parties du drone, à savoir les trains d'atterrissage et les ailes, par rapport à la visée de la charge utile, dont l'orientation et l'inclinaison changent régulièrement, indépendamment de la position et des mouvements du vecteur. L'information de la position de ces éléments permet à l'opérateur d'anticiper sur d'éventuels « décrochages » de

l'algorithme de suivi.

Afin de transmettre cette information à l'opérateur, nous pouvons imaginer d'ajouter sur la vue caméra des parties « fantômes » affichées en transparence sur les bords de l'écran en anticipation de l'apparition de la partie réelle dans le champ de la caméra, permettant à l'opérateur de se préparer au déverrouillage du suivi ou de prendre l'action d'activer le mode manuel.

Éléments propres à la tâche « Rechercher en IR »

Les éléments de situation associés uniquement à la tâche « Rechercher en IR » sont les suivants : *inclinaison de la charge utile (pitch)*, *orientation de la charge utile (yaw)*, *mode manuel*, *historique de la trace senseur*, *portée DRI*, *animal*.

Nous retrouvons les éléments propres au contrôle en manuel décrivant *l'orientation* et *l'inclinaison* de la CU, paramètres sur lesquels l'opérateur influe directement dans ce mode. L'élément *mode manuel* est saillant selon la définition de Stanton. La conscience du mode est une information importante lors d'un changement de mode, en particulier lorsque celui-ci n'a pas été déclenché par l'opérateur (que ce soit par une défaillance du système, comme dans notre exemple, ou par un choix de ce dernier, dans le cadre d'une politique de « partage d'autorité »).

Afin de retrouver la bonne visée, l'opérateur pourra prendre connaissance de *l'historique de la trace senseur*, et notamment observer quels étaient les dernières zones observées lors du suivi de la piste afin de retrouver plus rapidement une visée sur celle-ci, tâche qui peut être difficile lorsque le vecteur est en mouvement, et plus particulièrement en train d'effectuer un virage.

7.6.4.2 Éléments partagés et sous-graphes communs

Éléments partagés

Parmi les éléments partagés entre les contextes des deux tâches, certains sont indépendants de tout graphe : *nuage*, *mode de contrôle CU* et *trace senseur*.

Nous retrouvons l'information portant sur les *nuages*, déjà traitée dans la première analyse de transition (§ 7.6.3.2).

Le *mode de contrôle CU* est une information importante dans le cas d'un changement de mode, car lorsqu'elle est connue de l'opérateur, elle permet d'éviter la confusion sur le mode courant.

La *trace senseur* représentant la projection au sol de la visée de la charge utile est importante dans ces deux modes de contrôle mais également dans le cadre du pilotage du drone, en mode *Fly The Footprint*, qui peut se traduire par « Vole en suivant la trace au sol ». Si cette trace senseur est asservie à la position de la cible suivie avant la transition, et donc contrôle le vol du vecteur, ce ne sera plus le

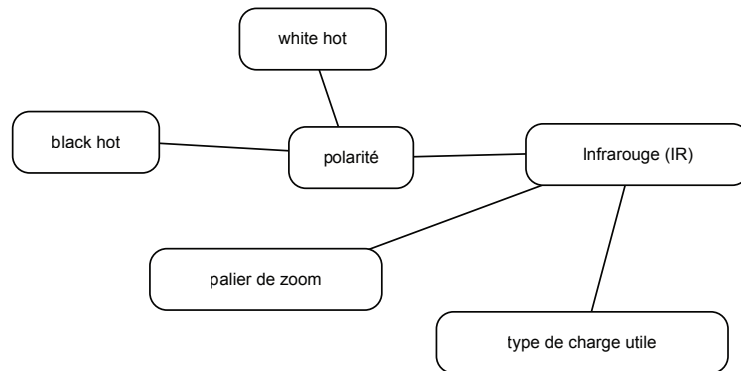


FIGURE 7.26 : partie du sous-graphe commun relatif à l'optique infrarouge.

cas après le changement de mode, où elle dépend des commandes d'orientation et d'inclinaison définies par l'opérateur.

Ainsi, dans le scénario considéré, lors du « décrochage » du suivi, la trace capteur ne se déplacera plus. Le drone arrêtera en conséquence son virage, augmentant la difficulté de récupération de la piste dans le champ de la caméra. Une solution pour fluidifier la transition serait d'introduire une assistance au contrôle de la charge utile par la trace capteur, directement sur l'interface cartographique, permettant à l'opérateur de s'appuyer sur l'*historique de la trace capteur*, afin de voir les dernières positions observées et pouvoir retrouver la piste en dirigeant la charge utile avec un dispositif de pointage. Dans le cas d'un contrôle de la charge utile au joystick, cette assistance pourrait se traduire par l'affichage d'une suite d'instructions de directions de déplacement de la CU, sur l'écran vidéo, afin de guider l'opérateur dans la manipulation et réduire l'écart de la trace au sol par rapport à l'historique.

Plus grands sous-graphes communs

L'analyse de cette transition fait ressortir un seul plus grand sous-graphe commun. Néanmoins, ce graphe peut être divisé en deux parties : une partie relative à l'optique infrarouge de la charge utile (figure 7.26), et une deuxième partie relative à la piste suivie (figure 7.27). Ces deux sous-graphes sont liés aux niveaux des éléments *chaleur* et *Infrarouge (IR)*.

Le premier sous-graphe (figure 7.26) porte donc sur les paramètres relatifs à l'optique Infrarouge. Ces informations sont partagées car il n'y a pas de changement d'optique lors du changement de mode de contrôle. Lors de la transition, il faut s'assurer de la continuité de ces différents paramètres : la polarité et le palier de zoom doivent être conservés afin de réduire le nombre de paramètres différents.

Comme lors de l'analyse de la première transition (§ 7.6.3.2), le sous-graphe relatif aux éléments décrivant la piste est partagé par les deux contextes. Néanmoins, dans le cas d'un changement de mode de contrôle, certaines de ces informations vont

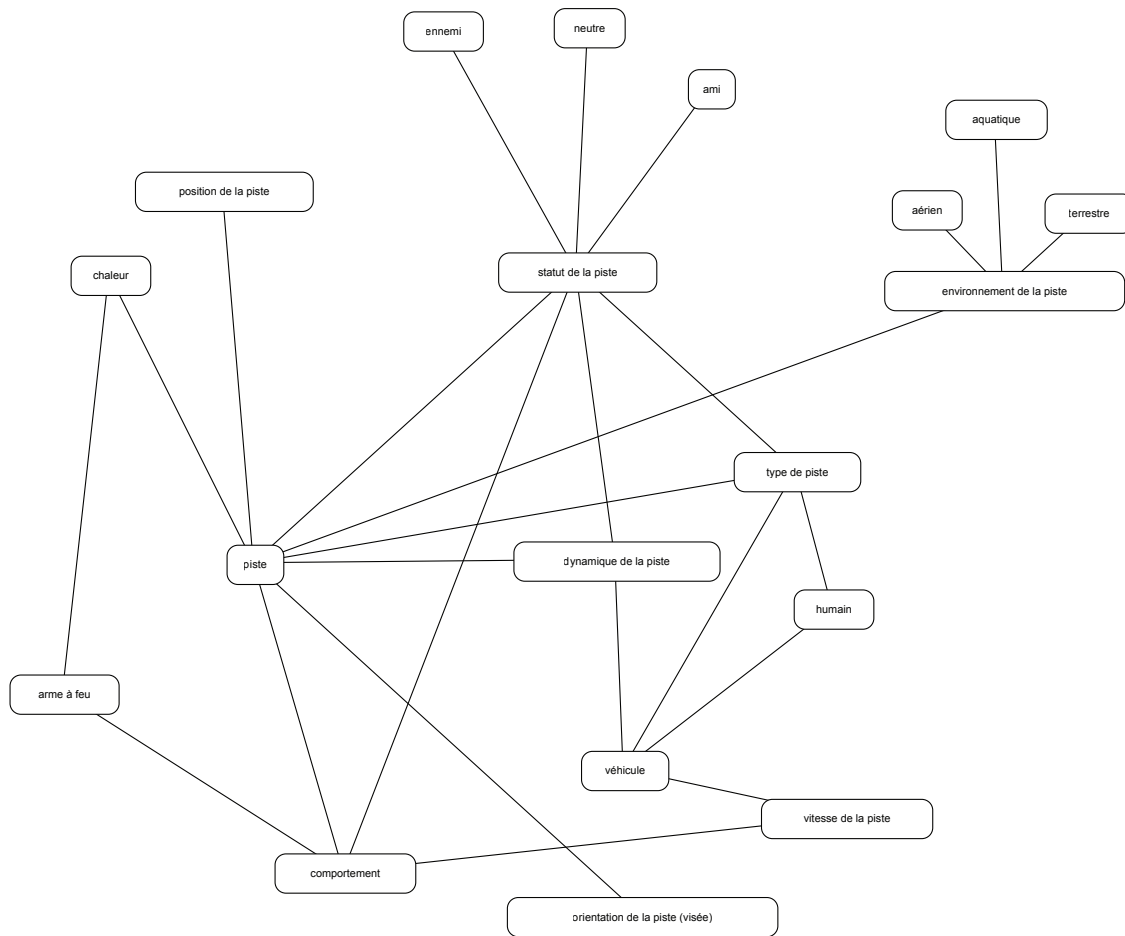


FIGURE 7.27 : partie du sous-graphe commun relative à la piste suivie.

devoir être renforcées afin de permettre à l'opérateur de retrouver un fonctionnement nominal et de récupérer au plus vite une information sur la piste.

Ainsi, on pourra s'appuyer sur la dernière position connue de la piste, associée à sa vitesse calculée par l'algorithme de suivi de piste et à sa direction afin d'extrapoler sa position future. Cette information pourra être affichée sur la cartographie, sous la forme d'un « fantôme » par exemple, à la manière des systèmes GPS de conduite, qui extrapolent la position du véhicule lors d'un passage sous un tunnel à partir des informations de vitesse et de direction prises en entrée du tunnel (principe de navigation à l'estime ou *dead reckoning*).

7.7 SYNTHÈSE DE L'ANALYSE DU CAS APPLICATIF

Les analyses présentées nous ont permis d'illustrer la mise en œuvre de la méthodologie SAMANTA au cas complexe de la conception d'un système de contrôle/commande de drone. Au cours de l'analyse des deux transitions considérées, issues d'un

scénario opérationnel, nous avons pu mettre en exergue les impacts que pouvaient avoir ces changements de tâche sur la *Situation Awareness* de l'opérateur et la façon dont nous pouvions intégrer des solutions d'interface pour diminuer cet impact, soit en préparant l'opérateur en amont de la transition, soit en lui permettant de reconstruire une SA optimale en aval.

Afin de compléter l'analyse de ces deux transitions, l'ergonome pourra par exemple y intégrer les tâches parallèles, que nous avons déjà mentionnées dans certaines parties de l'analyse. Les contextes associés à ces tâches viendront ajouter de nouvelles exigences en termes de *Situation Awareness*, ou encore renforcer celles mises en avant dans les analyses des transitions « atomiques », lorsqu'ils partagent des éléments de situation.

De la même façon, une réflexion pourra être menée sur la manière d'intégrer l'interruption de la tâche de prise de vue afin d'assurer la continuité de la tâche de contrôle de Charge Utile.

7.8 AUTRE APPLICATION DE LA MÉTHODOLOGIE SAMANTA

La méthodologie SAMANTA a également été appliquée au cas de la conduite d'une centrale nucléaire de type EPR (*European Pressurized water Reactor*). Cette application nous a permis de valider l'usage de la méthodologie et de l'outil associé pour l'analyse d'une transition dans un cas concret différent du domaine des drones. Elle a été réalisée sur la base de l'expérience d'une ergonome expert de Bertin Technologies dans le domaine du nucléaire et sur les informations rendues publiques pour la conception de l'EPR français par EDF⁴.

Dans cette section, nous présentons succinctement le cas d'étude considéré et détaillons les critiques émises par l'ergonome experte ayant réalisé cette étude.

Notons que cette étude fait l'objet d'une publication (Villaren *et al.*, 2012b).

7.8.1 Présentation du cas d'étude

Conduire une installation nucléaire de type EPR signifie « piloter de façon sûre et efficace l'installation nucléaire afin de répondre aux objectifs de production liés à la demande du réseau électrique » (Labarthe *et al.*, 2011). Ainsi, pour conduire une centrale de type EPR, les équipes d'opérateurs doivent être capables de gérer différents « modes » de contrôle : la conduite normale, la conduite incidentelle et accidentelle (CIA) et la conduite en accident grave.

De plus, en conduite normale, les opérateurs doivent assurer conjointement la production d'énergie et la maintenance préventive de l'installation. Concrètement, ceux-ci doivent réaliser (ou faire réaliser) « les contrôles, les essais, les entretiens, les

4. Rapport préliminaire de sûreté de Flamanville 3, mis à disposition par EDF sur <http://www.edf.com/html/epr/rps/index.pdf>.

réparations et les remplacements destinés à réduire la fréquence et l'incidence des défaillances d'un matériel » (*Rapport préliminaire de sûreté de Flamanville 3*). Le but de cette démarche est principalement d'optimiser les coûts directs de maintenance, tout en garantissant un objectif global de disponibilité de l'installation plus important, pour produire davantage d'énergie.

7.8.2 Démarche adoptée

Nous nous sommes intéressés à la problématique liée à la conduite normale associée à la réalisation d'essais périodiques pour répondre aux objectifs de maintenance préventive de l'installation nucléaire. Les tâches concernées par cette analyse se déroulent depuis la salle de commande principale et sont effectuées par l'équipe de conduite, composée de deux opérateurs et un superviseur. La transition considérée dans notre exemple consiste en un changement de tâches, dans le cadre de la conduite normale, entre la production d'énergie et la réalisation de la maintenance préventive sur un système spécifique (le DEL, un système de production et distribution d'eau glacée).

À l'issue des étapes de modélisation et de définition des contextes, réalisées à partir des connaissances de l'ergonome et des sources publiques d'EDF, l'ergonome a formulé des hypothèses portant sur les éléments en jeu pour la transition entre les tâches choisies (« Conduire la centrale EPR » vers « Réaliser un essai périodique »). Son expertise dans le domaine de la conduite de centrale lui a permis d'identifier en amont les éléments de situation estimés importants au moment de la transition, du fait de leur rôle informateur dans les deux tâches. L'ergonome a ainsi fait l'hypothèse que ces éléments devaient être identifiés comme « partagés » et « saillants » par l'outil lors de l'analyse.

7.8.3 Synthèse des commentaires

La prise en main de l'outil par l'ergonome a été rapide, malgré la présence de trois onglets différents, et la multiplication potentielle des manipulations à faire par onglet afin de lier les modèles existants. L'importation du modèle de tâches a demandé un minimum d'apprentissage de la part de l'ergonome. L'importation du modèle de SA s'est ensuite déroulée sans difficulté, sur la base de cet apprentissage.

La définition des contextes pour chaque tâche s'est également faite de façon naturelle, lorsque le principe a été compris (sélection de la tâche puis double-clic sur les éléments de situations à associer, voir section 6.4.2.3).

De plus, l'ergonome a apprécié l'effort effectué sur la représentation graphique du modèle de tâches proposée au sein de l'outil, jugée plus esthétique que celle de CTT.

Néanmoins, le fait que la construction des modèles de tâches et SA devait se faire au préalable de leur importation s'est avéré être une gêne potentielle. En effet,

la vérification de l'exhaustivité de la description des tâches, et surtout des éléments de situation, est réalisée au moment de l'étape de définition des contextes, où la présence (ou l'absence) d'éléments de situation représentant les connaissances mobilisées pour réaliser une tâche donnée peut être remarquée. Dans le cas d'étude choisi, très peu d'éléments se sont avérés manquants mais, dans des cas plus complexes, cela peut constituer un frein à l'utilisation de l'outil, nécessitant de revenir à la construction des modèles et à renouveler leur importation.

L'ergonome a apprécié la facilité de création d'une analyse par la sélection de deux tâches dans le modèle de tâches présenté. L'analyse automatique proposée par l'outil s'est révélée être en cohérence avec les hypothèses formulées *a priori* par l'ergonome, notamment pour les éléments devant être conservés tout au long de la transition et mis en évidence au moment du changement de tâches (Villaren *et al.*, 2012b).

Par ailleurs, l'ergonome a apprécié les informations supplémentaires fournies par l'outil d'analyse des transitions telle que la catégorisation proposée à l'issue de l'analyse des transitions, qui « apporte une information complémentaire à l'analyse *classique* en ergonomie », afin de définir, « pour chaque catégorie de transition, une solution permettant de mieux accompagner l'opérateur dans le maintien de sa conscience de situation » (Villaren *et al.*, 2012b).

7.8.4 Synthèse des recommandations de l'expert

L'outil conçu pour mettre en œuvre l'étape d'analyse de transitions de tâches de SAMANTA a été qualifié de « satisfaisant » lors de cette application. Néanmoins, des améliorations ont été proposées, afin de le rendre plus adapté aux besoins des ergonomes :

- Pour le *Modèle de Situation* : lors de l'importation du réseau proportionnel créé avec *yEd Graph Editor*, il serait intéressant de pouvoir récupérer la totalité des paramétrages disponibles dans ce logiciel, voire même de pouvoir choisir ceux qui sont importés (couleurs, formats des traits, etc.). La représentation visuelle de ce réseau d'informations en serait également améliorée.
- Pour les *Modèle de Tâches* et *Modèle de Situation* : il serait intéressant de simplifier le processus de mise à jour des informations importées dans l'outil, soit en autorisant leur modification directe après importation, soit en permettant d'importer à nouveau un fichier mis à jour et d'en intégrer les changements. Ceci permettrait de conserver les contextes déjà définis.
- Enfin, pour la fonction « Analyse » : au-delà de l'identification des différents sous-graphes présentés séparément, il serait intéressant de les visualiser dans l'ensemble du réseau proportionnel (par exemple, avec le même principe d'éléments coloriés dans la miniature). Ceci permettrait de pouvoir accéder rapidement à la localisation, dans le graphe global, des éléments communs aux deux

tâches.

L'ergonome a noté l'intérêt d'un tel outil pour la mise en application de la méthodologie à un cas d'étude plus complexe. En effet, dans l'étude présentée, réalisée sur une portion du système complexe considéré et pour une seule transition, il peut être envisagé de réaliser manuellement l'analyse à partir des modèles définis, mais l'analyse automatique mise en œuvre sur des cas plus complexes peut représenter un gain important de temps, de précision et d'exhaustivité.

8

Conclusion générale

Les travaux présentés dans ce mémoire de thèse concernent l'étude des changements de tâche des opérateurs de système complexes, afin de prendre en compte la dynamique de la situation induite dans la conception des interfaces de ces systèmes. L'approche adoptée s'appuie sur la construction d'un modèle des tâches et des contextes associés pour décrire l'activité de l'opérateur, permettant l'analyse amont des transitions et la détection de leur impact éventuel sur la *Situation Awareness* de l'opérateur. L'intégration de ces analyses se traduit par des recommandations portant sur l'interface des futures systèmes.

La démarche proposée se traduit par l'application d'une méthodologie de conception, mise en œuvre par un(e) ou plusieurs analyste(s) ayant des connaissances dans les domaines de l'ergonomie et des facteurs humains. Celle-ci intègre les utilisateurs finaux des systèmes considérés à plusieurs étapes clés et s'appuie sur ces données pour la construction des modèles.

8.1 RÉSUMÉ DES CONTRIBUTIONS

La première partie de ce mémoire présente une revue bibliographique structurée en deux chapitres.

Le chapitre 2 présente différentes approches, issues du domaine de l'Interaction Homme-Machine et de la Psychologie Cognitive, intégrant la dynamique temporelle dans la description des tâches ou leur analyse. Néanmoins, bien que centrés sur les utilisateurs finaux, les travaux présentés intègrent rarement la composante dynamique de la situation à laquelle ceux-ci sont confrontés.

En examinant le concept de *Situation Awareness*, le chapitre 3 cherche à déterminer de quelle manière ces aspects changeants de la situation tactique peuvent être modélisés, afin d'améliorer leur prise en compte dans un processus de conception. Les travaux présentés abordent les problématiques portant sur la connaissance de la situation tactique d'un point de vue global à la mission de l'opérateur, et ne traitent pas du cas spécifique de chaque tâche et des transitions entre ces tâches.

Ainsi, le constat effectué en bilan de cette première partie souligne l'absence de cadre de travail formalisé permettant d'unifier à la fois la dynamique temporelle des tâches exécutées par un opérateur de système complexe et la dynamique de la situation associée à sa mission.

8.1.1 Contributions conceptuelles

La troisième partie de ce mémoire présente les contributions conceptuelles de nos travaux.

Le chapitre 4 propose de pallier l'absence de lien entre les travaux décrivant la dynamique des tâches et les modèles de *Situation Awareness*.

A cette fin, une **taxinomie des transitions** a été introduite. Ce cadre d'étude conceptuel combine les deux axes *temporel* et *contextuel* et détermine quinze cas de figures prototypiques pour la description des transitions dans les systèmes complexes. Cette taxinomie décrivant les transitions au niveau des concepts est couplée à un **modèle dual des transitions**, associant le modèle de tâches ConcurTaskTree (Paternò *et al.*, 1997) et la description des contextes sous la forme de réseaux propositionnels d'éléments de situation, dans le but de concrétiser la taxinomie proposée.

Nous proposons deux usages possibles pour cette taxinomie associée au modèle dual des transitions.

- Le premier usage s'appuie sur la taxinomie et le modèle des transitions pour **prédire les impacts des transitions** sur l'opérateur, en s'appuyant notamment sur des travaux issus du domaine de la psychologie cognitive. En identifiant les transitions jugées difficiles, en fonction des impacts cognitifs pressentis, l'analyste peut ainsi cibler les transitions les plus pertinentes d'un scénario type, et approfondir leur analyse. Cet usage a été développé succinctement dans la deuxième moitié du chapitre 4, les considérations théoriques présentées constituent une perspective de cette thèse.
- Le deuxième usage vise à **aider la conception de nouveaux systèmes complexes** et constitue le développement du chapitre 5 et la contribution principale de cette thèse : il s'agit de la **méthodologie SAMANTA** (*Situation Awareness Modeling and ANalysis for Transition Amelioration*). Développée pour des utilisateurs ayant une forte sensibilité dans les domaines de l'ergonomie et des facteurs humains, cette méthodologie de conception s'appuie sur le modèle dual des transitions pour décrire l'activité d'un opérateur du système considéré. Elle intègre diverses méthodes d'analyse des transitions, afin de détecter leur impact sur la *Situation Awareness* et permettre à l'analyste de concevoir des interfaces adaptées.

8.1.2 Contributions pratiques

La troisième partie de ce mémoire présente les contributions pratiques de nos travaux.

Le chapitre 6 présente un outil développé afin d'appuyer une portion de la mise en œuvre de la méthodologie SAMANTA par les utilisateurs-ergonomes. Cet outil

se présente sous la forme d'une application Web dynamique et prend en charge les étapes de la méthodologie ne s'appuyant pas sur des outils ou modèles existants. Il s'inscrit ainsi dans une logique de « chaîne outillée », permettant à l'analyste de modéliser le système considéré puis de déterminer l'impact que peuvent avoir ces transitions sur l'opérateur.

La méthodologie SAMANTA et l'outil associé ont été mise en œuvre dans trois cas d'application.

- Un jeu vidéo de tir, Battlefield 3, dont l'étude offre une première illustration de la méthodologie SAMANTA (section 5.5).
- La conception d'un système de commande/contrôle de drone, qui a été modélisé et analysé à partir de données opérationnelles, issues d'un projet industriel. Ce cas d'étude est développé dans le chapitre 7.
- La conduite de centrale nucléaire, appliquée au réacteur EPR, qui a permis de valider l'application de la méthodologie à travers l'outil associé par une ergonome experte dans le domaine du nucléaire (section 7.8).

8.2 PERSPECTIVES

En conclusion de ce mémoire, nous présentons quelques perspectives envisagées et s'inscrivant dans la continuité des travaux de cette thèse.

8

8.2.1 Perspectives concernant la taxinomie

L'usage de la taxinomie en tant que moyen de prédiction de la « difficulté cognitive » des transitions a été abordé au chapitre 4 (section 4.3.2).

Afin de soutenir cet usage, des travaux supplémentaires doivent être menés. Le but serait de qualifier l'impact potentiel sur l'opérateur de chaque transition de la taxinomie en fonction de sa catégorie. Cette qualification pourrait être utilisée en amont de l'étude approfondie de certaines transitions, afin de filtrer celles qui, *a priori*, ne seront pas nuisibles à l'opérateur.

Dans ce but, certains travaux relatifs au *Task Switching*, présentés dans la section 2.2, pourront être approfondis et intégrés à l'étude de concepts plus généraux propres au domaine de la psychologie cognitive, comme les effets d'interférence en mémoire de nouveaux éléments ou d'éléments communs.

8.2.2 Perspectives concernant l'outil

L'outil développé dans le cadre de cette thèse constitue une première version vouée à évoluer en fonction des besoins exprimés par les utilisateurs-ergonomes. Certaines de ces évolutions ont été présentées en section 6.5 et en synthèse de la validation experte présentée dans la section 7.8.

D'autres évolutions, ayant une perspective à plus long terme, sont envisagées pour cet outil. En particulier, il pourrait être intéressant de générer un document faisant un bilan de l'analyse des transitions sélectionnées, et permettant de visualiser les données impliquées dans ces analyses, afin de faciliter le travail de l'ergonome.

L'utilisation de l'outil par les ergonomes étant l'objectif premier, ceci nécessite de prendre en compte en priorité leurs retours, la considération de développements supplémentaires venant s'inscrire en second plan.

Enfin, la validation experte réalisée dans le cadre de cette thèse présente une première étape de validation de l'outil mais un processus de validation à plus grand échelle, basé sur des données à la fois qualitatives et quantitatives, pourrait être envisagée, permettant à la fois de valider l'utilité de la méthodologie et l'adéquation de l'outil à celle-ci.

8.2.3 Perspectives concernant la méthodologie

La méthodologie SAMANTA a été développée afin de soutenir l'effort de conception de nouveaux systèmes complexes en s'appuyant sur l'analyse de systèmes existants et l'intégration de besoins futurs des utilisateurs. Nous envisageons plusieurs évolutions possibles (et compatibles) pour cette méthodologie.

8.2.3.1 Utilisation pour l'évaluation de systèmes existants

Le point de vue développé jusqu'à présent est d'utiliser la méthodologie SAMANTA *en amont* de la conception des systèmes complexes, afin d'appuyer ce processus. Nous pensons que cette méthodologie peut également être utilisée *a posteriori*, pour l'évaluation de systèmes complexes existants, afin de mettre en exergue les transitions difficiles de tels systèmes. Ainsi, l'analyste pourra dérouler les mêmes étapes, modélisant le système à partir des données issues des opérateurs et en définissant les contextes de la même façon.

Afin d'évaluer les transitions, les résultats des différentes analyses seront comparés aux interfaces existantes. L'analyste pourra ensuite vérifier quelles sont les exigences satisfaites en termes de SA et juger si la transition considérée est effectivement soutenue par l'interface. Dans le cas contraire, il pourra proposer des améliorations de l'interface afin de tenir compte de ces transitions.

De la même manière qu'en phase de conception, l'application de la méthodologie pour l'évaluation peut rapidement devenir fastidieuse, du fait du nombre de tâches à décrire. Afin de guider le choix des transitions à analyser, l'analyste pourra donc s'appuyer sur les retours des opérateurs, notamment s'ils mettent en avant des transitions jugées difficiles lors des entretiens, en plus de la définition de scénarios opérationnels types.

8.2.3.2 Couplage de la méthodologie avec la formation des opérateurs

L'analyse des transitions réalisée au cœur de la méthodologie SAMANTA permet de déterminer quelles sont les transitions difficiles et de mettre en exergue les éléments devant être soutenus par l'interface. Dans les cas où l'interface ne suffit pas à soutenir seule ces transitions, le développement d'une formation spécifique des opérateurs à ces transitions, s'appuyant sur les résultats de l'analyse, pourrait permettre de cibler précisément les cas de figure difficiles afin d'y remédier par l'apprentissage, et ainsi diminuer le risque d'erreur.

Dans leurs travaux, Martinie *et al.* (2011) préconisent d'intégrer la définition des programmes de formations des opérateurs directement lors de la conception de systèmes interactifs critiques. Cette approche se traduit par l'utilisation conjointe de modèles des tâches, des modèles formels du comportement du système et du modèle de développement du programme de formation. Le modèle de tâches utilisé, HAMSTERS, étant basé sur les mêmes identificateurs LOTOS que ceux intégrés à CTT, nous pouvons imaginer d'adapter la démarche mise en œuvre afin d'y ajouter la prise en compte de la *Situation Awareness* en phase de conception. L'ajout de ce modèle de SA et la mise en œuvre de la méthodologie SAMANTA pour déterminer les points forts de la formation à développer pourraient permettre de renforcer la fiabilité de tels systèmes.

8.2.3.3 Intégration de l'évolution temporelle des contextes

Les travaux présentés proposent diverses méthodes d'analyse des transitions. Ces méthodes s'appuient principalement sur l'analyse des graphes décrivant chaque contexte. L'aspect temporel des transitions n'est ainsi pris en considération qu'à travers l'étude des contextes *avant* et *après* la transition.

Néanmoins, dans certains cas, lors d'un changement de tâches, il peut être intéressant d'étudier l'évolution des contextes et l'instant « d'apparition » ou de « disparition » des différents éléments des graphes associés. Nous avons abordé les transitions comme le passage discret d'un état à un autre, mais ce passage peut être considéré comme une évolution continue des éléments de la situation, passant par différents contextes intermédiaires dont l'étude peut être intéressante (principe illustré par la figure 8.1). D'un point de vue cognitif, nous pouvons imaginer que, dans certains cas, l'activation des éléments en mémoire suit un schéma prédéterminé qu'il pourrait être intéressant de copier à travers l'interface. Ainsi, ce principe de diffusion de l'activation (Kekenbosch et Denhière, 1988) pourrait se traduire par le changement progressif des éléments de l'interface, rendant disponibles certains éléments avant d'autres afin d'accompagner le « chargement » d'un schéma en mémoire.

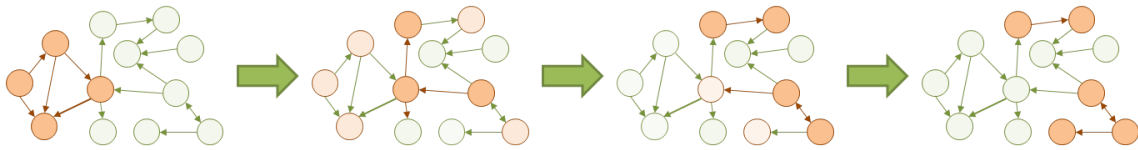


FIGURE 8.1 : illustration de l'évolution de l'activation des éléments de situation lors d'une transition entre tâches.

D'autres méthodes d'analyse pourront également être considérées, notamment en s'appuyant sur les données de CTT non prises en compte, telles que les opérateurs de planification locale (durée des tâches, fréquence, priorité, etc.). Par exemple, la fréquence de la tâche peut avoir un impact plus ou moins important sur l'interface, selon son effet escompté. La mise en œuvre de méthodes d'extraction de données ou issues du monde de la fouille de données pourrait également être étudiée.

Enfin, il peut être intéressant de tirer partie du caractère « orienté » du réseau propositionnel et de son lien avec la psychologie cognitive. En effet, ce réseau représente des cheminements de pensée de l'opérateur qui, lors d'une transition, pourraient avoir une signification particulière.

8

8.2.3.4 Intégration à une démarche de l'Ingénierie Dirigée par les Modèles

Enfin, une perspective d'évolution porte sur l'intégration de la méthodologie à une démarche de conception issue de l'Ingénierie Dirigée par les Modèles (*Model-Driven Engineering*).

La conception d'interface dans le domaine de l'Ingénierie Dirigée par les Modèles (IDM) s'appuie sur la décomposition de la description de l'interface utilisateur en plusieurs niveaux d'abstraction, permettant au concepteur d'adapter une même interface à de « multiples appareils, plateformes, contextes, utilisateurs, modalités, cultures et organisations »¹.

Le cadre conceptuel *Cameleon* (Calvary *et al.*, 2003) détermine ainsi quatre niveaux d'abstraction pour la description des interfaces :

- Le niveau **Tâches & Concepts** décrit les tâches exécutées par les utilisateurs et les concepts du domaine pris en compte dans le système.
- Le niveau de l'**interface utilisateur abstraite** (*Abstract UI*, AUI) décrit les espaces interaction, indépendamment de la modalité utilisée, en groupant les sous-tâches dans des *objets d'interaction abstraits*.
- Le niveau de l'**interface utilisateur concrète** (*Concrete UI*, CUI) définit l'agencement et les principes de navigation entre les éléments de l'interface. Ce niveau définit également l'aspect de l'interface (*look and feel*) sous la forme de maquettes par exemple.
- Le niveau de l'**interface utilisateur finale** (*Final UI*, FUI) présente une version de l'interface spécifique à une plateforme, dans un langage de program-

1. concept du $\mu 7$ décrit par le projet <http://usixml.eu/>

mation défini et dépendant de la plateforme sur laquelle elle sera déployée.

La méthodologie SAMANTA peut s'insérer dans un *framework* de ce type. Les modèles construits dans le cadre de l'étape 2 constituent le niveau « Tâches & Concepts » de la description et les résultats des analyses menées peuvent s'exprimer aux différents niveaux d'abstraction de cette approche. Les impacts sur l'IHM seraient traduits sous la forme de recommandations générales sur l'agencement ou l'interaction, mais aussi à des niveaux plus précis comme le contenu de chaque zone de l'interface.

L'approche $\mu 7$ prônée par le projet UsiXML (s'appuyant sur le cadre conceptuel *Cameleon*) possède un champ d'application plus large que celui de la méthodologie SAMANTA, notamment en termes d'adaptation à différentes plateformes et différents appareils. Cependant, la formalisation du système complexe considéré et des recommandations issues des analyses, dans un langage de description des interfaces utilisateurs (UIDL) tel que UsiXML (Limbourg *et al.*, 2004) peut faciliter le travail de conception itératif généralement mis en place. L'usage de tel langage permet de soutenir la modularité du système et assure sa pérennité en permettant d'éventuelles modifications ultérieures.

Publications

Villaren, Thomas, Gilles Coppin et Angélica Leal (2012). ***Modeling Task Transitions to Help Designing for Better Situation Awareness***. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems* (EICS '12). Copenhagen, Danemark. pp. 195–204.

Villaren, Thomas, Angélica Leal et Gilles Coppin (2012). **Contribution de la méthodologie SAMANTA à l'ergonomie d'interfaces en support des changements de tâche**. In Actes de la 24^{ème} Conférence francophone sur l'Interaction Homme-Machine (ErgoIHM'12). Biarritz, France.

Villaren, Thomas, Charlie Madier, François Legras, Angélica Leal, Brice Kovács et Gilles Coppin (2010). ***Towards a Method for Context-Dependent Allocation of Functions***. In *Proceeding of the 2nd Conference on Humans Operating Unmanned Systems* (HUMOUS'10). ONERA/ISAE, Toulouse, France.



Bibliographie

- ABED, M., TABARY, D. et KOLSKI, C. (2003). Using Formal Specification Techniques for the Modeling of Tasks and the Generation of Human-Computer User Interface Specifications. *In The Handbook of Task Analysis for Human-Computer Interaction*, chapitre 25, pages 503–529. Lawrence Erlbaum Associates.
- ADAMCZYK, P. D. et BAILEY, B. P. (2004). If Not Now, When?: The Effects of Interruption at Different Moments Within Task Execution. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*, volume 6, pages 271–278. ACM Press.
- ALLEN, J. F. (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843.
- ALLPORT, A., STYLES, E. A. et HSIEH, S. (1994). Shifting Intentional Set: Exploring the Dynamic Control of Tasks. *In UMILTÀ, C., éditeur : Attention and Performance XV: Conscious and Nonconscious Information Processing*, volume XV, chapitre 17, pages 421–452. The MIT Press.
- ALLPORT, A. et WYLIE, G. (2000). Task Switching, Stimulus-Response Bindings, and Negative Priming. *In MONSELL, S. et DRIVER, J., éditeurs : Control of Cognitive Processes: Attention and Performance XVIII*, chapitre 2, pages 35–70. The MIT Press.
- ALTMANN, E. M. et GRAY, W. D. (2008). An Integrated Model of Cognitive Control in Task Switching. *Psychological Review*, 115(3):602–639.
- ALTMANN, E. M. et TRAFTON, J. G. (2002). Memory for goals: an activation-based model. *Cognitive Science*, 26(1):39–83.
- ALTMANN, E. M. et TRAFTON, J. G. (2004). Task Interruption: Resumption Lag and the Role of Cues. *In FORBUS, K., GENTNER, D. et REGIER, T., éditeurs : Proceedings of the 26th Annual Meeting of the Cognitive Science Society*, pages 42–47.
- AMALBERTI, R. (1996). *La conduite des systèmes à risque*. Presses Universitaires de France.
- ANDERSON, J. R. (1989). A Theory of the Origins of Human Knowledge. *Artificial Intelligence*, 40(1-3):313–351.

- ANDERSON, J. R. et LEBIERE, C. (1998). *The Atomic Components of Thought*. Lawrence Erlbaum Associates.
- ANNETT, J. et DUNCAN, K. D. (1967). Task Analysis and Training Design. Rapport technique, Hull University - Department of Psychology.
- ARRINGTON, C. M. et LOGAN, G. D. (2004). The Cost of a Voluntary Task Switch. *Psychological Science*, 15(9):610–615.
- ARRINGTON, C. M., LOGAN, G. D. et SCHNEIDER, D. W. (2007). Separating Cue Encoding From Target Processing in the Explicit Task-Cuing Procedure: Are There "True" Task Switch Effects? *Journal of Experimental Psychology. Learning, Memory, and Cognition*, 33(3):484–502.
- ARTMAN, H. et GARBIS, C. (1998). Situation Awareness as Distributed Cognition. In GREEN, T., BANNON, L., WARREN, C. et BUCKLEY, J., éditeurs : *Cognition and Cooperation. Proceedings of the 9th Conference of Cognitive Ergonomics*, pages 151–156, Limerick: Ireland.
- BAILEY, B. P. et IQBAL, S. T. (2008). Understanding Changes in Mental Workload during Execution of Goal-Directed Tasks and Its Application for Interruption Management. *ACM Transactions on Computer-Human Interaction*, 14(4):28.
- BAILEY, B. P. et KONSTAN, J. A. (2006). On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in Human Behavior*, 22(4):685–708.
- BAILLARD, C. et MAÎTRE, H. (1999). 3-D Reconstruction of Urban Scenes from Aerial Stereo Imagery: A Focusing Strategy. *Computer Vision and Image Understanding*, 76(3):244–258.
- BALBO, S., OZKAN, N. et PARIS, C. (2003). Choosing the Right Task-Modeling Notation: A Taxonomy. In DIAPER, D. et STANTON, N. A., éditeurs : *The Handbook of Task Analysis for Human-Computer Interaction*, chapitre 22, pages 445–465. Lawrence Erlbaum Associates.
- BASTIEN, J. M. C. et SCAPIN, D. L. (1993). Critères ergonomiques pour l'évaluation d'interfaces utilisateurs (version 2.1). Rapport technique, INRIA.
- BENGOETXEA, E. (2002). *Inexact Graph Matching Using Estimation of Distribution Algorithms*. Thèse de doctorat, École Nationale Supérieure des Télécommunications.
- BILLINGS, C. (1997). *Aviation Automation: the Search for a Human-Centered Approach*. Lawrence Erlbaum Associates.

BIBLIOGRAPHIE

- BOEHM-DAVIS, D. A. et REMINGTON, R. (2009). Reducing the disruptive effects of interruption: A cognitive framework for analysing the costs and benefits of intervention strategies. *Accident Analysis and Prevention*, 41(5):1124–1129.
- BOLSTAD, C. A., CUEVAS, H. M., GONZALEZ, C. et SCHNEIDER, M. (2005). Modeling Shared Situation Awareness. In *Proceedings of the 14th Conference on Behavior Representation In Modeling and Simulation (BRIMS)*.
- BRASS, M., RUGE, H., MEIRAN, N., RUBIN, O., KOCH, I., ZYSSET, S., PRINZ, W. et von CRAMON, D. Y. (2003). When the same response has different meanings: Recoding the response meaning in the lateral prefrontal cortex. *NeuroImage*, 20(2):1026–1031.
- BRETON, R. et ROUSSEAU, R. (2003). Situation Awareness: A review of the concept and its measurement. Rapport technique, DRDC Valcartier TR.
- BRUNSWIK, E. (1956). *Perception and the representative design of psychological experiments*. Berkeley: University of California Press.
- BRYANT, D. J., LICHACZ, F. M. J., HOLLANDS, J. G. et BARANSKI, J. V. (2004). Modeling Situation Awareness in an Organisational Context: Military Command and Control. In BANBURY, S. et TREMBLAY, S., éditeurs : *A Cognitive Approach to Situation Awareness: Theory and Application*, chapitre 6, pages 104–116. Ashgate Pub Ltd.
- CADES, D. M., BOEHM-DAVIS, D. A., TRAFTON, J. G. et MONK, C. A. (2007). Does the Difficulty of an Interruption Affect our Ability to Resume? *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 51(4):234–238.
- CADES, D. M., TRAFTON, J. G. et BOEHM-DAVIS, D. A. (2006). Mitigating Disruptions: Can Resuming an Interrupted Task Be Trained? *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(3):368–371.
- CAFFIAU, S., SCAPIN, D. L., GIRARD, P., BARON, M. et JAMBON, F. (2010). Increasing the expressive power of task analysis: Systematic comparison and empirical assessment of tool-supported task models. *Interacting with Computers*, 22(6):569–593.
- CAFFIAU, S., SCAPIN, D. L. et SANOU, L. (2008). Retour d'Expérience en Enseignement de la Modélisation de Tâches. In *Proceedings of ErgoIA'08*, pages 135–143.
- CALVARY, G., COUTAZ, J., THEVENIN, D., LIMBOURG, Q., BOUILLON, L. et VANDERDONCKT, J. (2003). A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15(3):289–308.

- CARD, S., MORAN, T. et NEWELL, A. (1983). *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates.
- CHANQUOY, L., TRICOT, A. et SWELLER, J. (2007). *La charge cognitive*. Armand Collin - Collection U.
- COOPER, R. et SHALLICE, T. (2000). Contention Scheduling and the Control of Routine Activities. *Cognitive Neuropsychology*, 17(4):297–338.
- CRANDALL, B., KLEIN, G. A. et HOFFMAN, R. R. (2006). *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*. The MIT Press.
- CUMMINGS, M. L., HOW, J. P., WHITTEN, A. et TOUPET, O. (2012a). The Impact of Human-Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control. *Proceedings of the IEEE*, 100(3):660–671.
- CUMMINGS, M. L., TAPPAN, J. et MIKKELSEN, C. (2012b). One Work Analysis, Two Domains: A Display Information Requirements Case Study. *In Proceedings of the 56th Annual Meeting of the Human Factors and Ergonomic Society (HFES)*.
- CZERWINSKI, M., CHRISMAN, S. et SCHUMACHER, B. (1991). The effects of warnings and display similarities on interruptions in multitasking environments. *ACM SIGCHI Bulletin*, 23(4):38–39.
- CZERWINSKI, M., CUTRELL, E. et HORVITZ, E. (2000). Instant Messaging and Interruption: Influence of Task Type on Performance. *In Proceedings of OZCHI*, pages 356–361. ACM Press.
- DE JONG, R. (2000). An Intention-Activation Account of Residual Switch Costs. *In MONSELL, S. et DRIVER, J., éditeurs : Control of Cognitive Processes: Attention and Performance XVIII*, chapitre 15, pages 357–376. MIT Press.
- DESSART, C.-E., GENARO MOTTI, V. et VANDERDONCKT, J. (2011). Showing User Interface Adaptivity by Animated Transitions. *In Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems (EICS'11)*, pages 95–104. ACM Press.
- DIAPER, D. et STANTON, N. A. (2004). *The Handbook of Task Analysis for Human-Computer Interaction*. Lawrence Erlbaum Associates.
- DISMUKES, K., YOUNG, G. et SUMWALT, R. (1998). Cockpit Interruptions and Distractions. *ASRS Directive*, (10):4–9.
- DoD (2012). Department of Defense Design Criteria Standard - MIL-STD-1472G. Rapport technique.

BIBLIOGRAPHIE

- DODHIA, R. M. et DISMUKES, K. (2003a). A Task Interrupted Becomes a Prospective Memory Task: Encoding and Retrieval Manipulations. *Poster presented at the 44th Annual Meeting of the Psychonomic Society.*
- DODHIA, R. M. et DISMUKES, K. (2003b). A Task Interrupted Becomes A Prospective Task. *Poster presented at the 15th Annual Convention of the American Psychological Society.*
- DOMINGUEZ, C. (1994). Can SA be defined? In VIDULICH, M. A., DOMINGUEZ, C., VOGEL, E. et McMILLAN, G., éditeurs : *Situation Awareness: Papers and Annotated Bibliography*, pages 5–15.
- DRAPER, M. H., CALHOUN, G. L., RUFF, H., MULLINS, B., LEFEBVRE, A., AYALA, A. et WRIGHT, N. (2008). Transition Display Aid for Changing Camera Views in UAV Operations. In *Proceedings of the 1st Conference on Human Operating Unmanned Systems (HUMOUS'08)*, pages 67–78, Brest, France. Télécom Bretagne.
- DURSO, F. T. et DATTEL, A. R. (2004). SPAM: The Real-Time Assessment of SA. In BANBURY, S. et TREMBLAY, S., éditeurs : *A Cognitive Approach to Situation Awareness: Theory and Application*, chapitre 8, pages 137–154. Ashgate Pub Ltd.
- DURSO, F. T., HACKWORTH, C. A., TRUITT, T. R., CRUTCHFIELD, J., NIKOLIC, D. et MANNING, C. A. (1998). Situation Awareness as a Predictor of Performance for En Route Air Traffic Controllers. *Air Traffic Control Quarterly*, 6(1):1–20.
- DURSO, F. T., TRUITT, T. R., HACKWORTH, C. A., CRUTCHFIELD, J., NIKOLIC, D., MOERTL, P. M., OHRT, D. et MANNING, C. A. (1995). Expertise and Chess: A Pilot Study Comparing Situation Awareness Methodologies. In GARLAND, D. J. et ENDSLEY, M. R., éditeurs : *Experimental Analysis and Measurement of Situation Awareness*, pages 295–301.
- EDWARDS, M. B. et GRONLUND, S. D. (1998). Task Interruption and its Effects on Memory. *Memory*, 6(6):665–687.
- ENDSLEY, M. R. (1993). A Survey of Situation Awareness Requirements in Air-to-Air Combat Fighters. *The International Journal of Aviation Psychology*, 3(2):157–168.
- ENDSLEY, M. R. (1995a). Measurement of Situation Awareness in Dynamic Systems. *Human Factors*, 37(1):65–84.
- ENDSLEY, M. R. (1995b). Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors*, 37(1):32–64.
- ENDSLEY, M. R. (2000a). Direct Measurement of Situation Awareness: Validity and Use of SAGAT. In ENDSLEY, M. R. et GARLAND, D. J., éditeurs : *Situation Awareness Analysis and Measurement*, chapitre 7, pages 147–174. Lawrence Erlbaum Associates.

- ENDSLEY, M. R. (2000b). Situation Models: An Avenue to the Modeling of Mental Models. *In Proceedings of the 44th Annual Meeting of the Human Factors and Ergonomics Society (HFES)*.
- ENDSLEY, M. R. (2000c). Theoretical Underpinnings of Situation Awareness: A Critical Review. *In* ENDSLEY, M. R. et GARLAND, D. J., éditeurs : *Situation Awareness Analysis and Measurement*, volume pp, chapitre 1, pages 3–32. Lawrence Erlbaum Associates.
- ENDSLEY, M. R., BOLTÉ, B. et JONES, D. G. (2003). *Designing for Situation Awareness: an Approach to User-Centered Design*. Taylor & Francis.
- ENDSLEY, M. R., FARLEY, T. C., JONES, W. M., MIDKIFF, A. H. et HANSMAN, R. J. (1998a). Situation Awareness Information Requirements for Commercial Airline Pilots. Rapport technique, MIT International Center for Air Transportation (ICAT).
- ENDSLEY, M. R., HOLDER, L. D., LEIBRECHT, B. C., GARLAND, D. J., WAMPLER, R. L. et MATTHEWS, M. D. (2000a). Modeling and Measuring Situation Awareness in the Infantry Operational Environment. Rapport de recherche, US Army Research Institute for the Behavioral and Social Sciences.
- ENDSLEY, M. R. et JONES, W. M. (1997). Situation Awareness Information Dominance & Information Warfare. Rapport technique 3, United States Air Force Armstrong Laboratory.
- ENDSLEY, M. R. et RODGERS, M. D. (1994). Situation Awareness Information Requirements Analysis for En Route Air Traffic Control. Rapport technique, US Department of Transportation – Federal Aviation Administration.
- ENDSLEY, M. R., SELCON, S. J., HARDIMAN, T. D. et CROFT, D. G. (1998b). A Comparative Analysis of SAGAT and SART for Evaluations of Situation Awareness. *In Proceedings of the 42th Annual Meeting of the Human Factors and Ergonomics Society (HFES)*, volume 42, pages 82–86.
- ENDSLEY, M. R., SOLLENBERGER, R. et STEIN, E. (2000b). Situation Awareness: A Comparison of Measures. *In Proceedings of the Human Performance, Situation Awareness and Automation: User Centered Design for the New Millenium Conference*.
- FAA (1996). Human Factors Design Guide (HFDG) for Acquisition of Commercial-off-the-Shelf (COTS) Subsystems, Non-Developmental Items (NDI), and Developmental Systems. Rapport technique, US Department of Transportation - Federal Aviation Administration.

BIBLIOGRAPHIE

- FELLER, A. et RUCKER, R. (1989). Optimization of Manufacturing Systems Design. In SHUNK, D. L., éditeur : *Proceedings of the IFIP WG 5.3 International Conference on Modeling and Simulation for Optimization of Manufacturing Systems Design and Application*, pages 171–194, Tempe, AZ, USA.
- FOURNIER, J.-C. (2006). *Théorie des graphes et applications*. Hermes Science, édition Lavoisier.
- FRANKE, J. L., DANIELS, J. J. et MCFARLANE, D. C. (2002). Recovering Context After Interruption. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society*, pages 310–315.
- GERTLER, J. (2012). U.S. Unmanned Aerial Systems. Rapport technique, Congress Research Service.
- GIBSON, J. J. (1979). *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates.
- GIESE, M., MISTRZYK, T., PFAU, A., SZWILLUS, G. et von DETTEN, M. (2008). AMBOSS: A Task Modeling Approach for Safety-Critical Systems. *Engineering Interactive Systems. Lecture Notes in Computer Science*, 5247:98–109.
- GIEVSKA, S., LINDEMAN, R. et SIBERT, J. (2005). Examining the Qualitative Gains of Mediating Human Interruptions during HCI. In *Proceedings of the 11th International Conference on Human-Computer Interaction*, Las Vegas, NV.
- GIEVSKA, S. et SIBERT, J. (2004). A Framework for Context-Sensitive Coordination of Human Interruptions in Human-Computer Interaction. *User-Centered Interaction Paradigms for Universal Access in the Information Society. Lecture Notes in Computer Science*, 3196:418–425.
- GIEVSKA, S. et SIBERT, J. (2005). Using Task Context Variables for Selecting the Best Timing for Interrupting Users. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, pages 171–176. ACM Press.
- GILLIE, T. et BROADBENT, D. (1989). What makes interruptions disruptive? A study of length, similarity, and complexity. *Psychological Research*, 50(4):243–250.
- GO, K. et CARROLL, J. M. (2003). Scenario-Based Task Analysis. In DIAPER, D. et STANTON, N. A., éditeurs : *The Handbook of Task Analysis for Human-Computer Interaction*, chapitre 5, pages 117–134. Lawrence Erlbaum Associates.
- GODBOLE, A. et SMARI, W. W. (2006a). A Methodology and Design Process for System Generated User Interruption based on Context, Preferences, and Situation

- Awareness. In *IEEE International Conference on Information Reuse & Integration*, pages 608–616.
- GODBOLE, A. et SMARI, W. W. (2006b). Human Perspective Based Context Acquisition, Learning and Awareness in the Design of Context Aware Systems. In *IEEE Military Communications Conference*, pages 1–8.
- GONZALEZ, C. (1996). Does Animation in User Interfaces Improve Decision Making? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'96)*, pages 27–34. ACM Press.
- GORE, B. F. (2010). Man-machine Integration Design and Analysis System (MIDAS) v5: Augmentations, Motivations, and Directions for Aeronautics Applications. In *Human Modeling in Assisted Transportation*. Keynote Lecture.
- GRAY, P. D., ENGLAND, D. et MCGOWAN, S. (1994). XUAN: Enhancing UAN to Capture Temporal Relationship among Actions. In COCKTON, G., DRAPER, S. et WEIR, G. R. S., éditeurs : *People and Computers IX: Proceedings of HCI'94*, pages 301–312. Cambridge University Press.
- HART, S. G. et STAVELAND, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In HANCOCK, P. A. et MESHKATI, N., éditeurs : *Human Mental Workload.*, pages 139–183. Elsevier.
- HARTSON, H. R. et GRAY, P. D. (1992). Temporal Aspects of Tasks in the User Action Notation. *Human Computer Interaction*, 7(1):1–45.
- HOC, J.-M. (1996). *Supervision et contrôle de processus. La cognition en situation dynamique*. Presses Universitaires de Grenoble.
- HODGETTS, H. M. et JONES, D. M. (2005). Interrupting Problem Solving: Effects of Interruption Position and Complexity. In *Past reflections, future directions: proceedings of the 40th Australian Psychological Society Annual Conference*, pages 128–132.
- HODGETTS, H. M. et JONES, D. M. (2006). Interruption of the Tower of London Task: Support for a Goal-Activation Approach. *Journal of Experimental Psychology: General*, 135(1):103–15.
- HOOEY, B. L., GORE, B. F., WICKENS, C. D., SCOTT-NASH, S., SOCASH, C., SALUD, E. et FOYLE, D. C. (2011). Modeling Pilot Situation Assessment. In CACCIABUE, P. C., HJÄLMDAHL, M., LUEDTKE, A. et RICCIOLI, C., éditeurs : *Proceeding of the Human Modeling in Assisted Transportation Conference*, pages 207–213.
- HUTCHINS, E. (1995). *Cognition in the Wild*. MIT Press.

BIBLIOGRAPHIE

- IQBAL, S. T. et BAILEY, B. P. (2005). Investigating the Effectiveness of Mental Workload as a Predictor of Opportune Moments for Interruption. *In Human Factors in Computing Systems, Proceedings of CHI'05: Extended Abstracts*, pages 1489–1492.
- IQBAL, S. T. et BAILEY, B. P. (2010). Oasis: A Framework for Linking Notification Delivery to the Perceptual Structure of Goal-Directed Tasks. *ACM Transactions on Computer-Human Interaction*, 17(4):15:1–28.
- ISO (1989). LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour.
- JAMBON, F. (1996). *Erreurs et interruptions du point de vue de l'ingénierie de l'Interaction Homme-Machine*. Thèse de doctorat, Université J. Fourier de Grenoble.
- JEANNOT, E., KELLY, C. et THOMPSON, D. (2003). The Development of Situation Awareness Measures in ATM Systems. Rapport technique, EATMP Infocentre.
- JERSILD, A. T. (1927). Mental Set and Shift. *Archives of Psychology*, 14(89).
- JOHN, B. E. et KIERAS, D. E. (1996). The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast. *ACM Transactions on Computer-Human Interaction (ToCHI)*, 3(4):320–351.
- JOHNSON, P., JOHNSON, H., WADDINGTON, R. et SHOULS, A. (1988). Task-Related Knowledge Structures: Analysis, Modelling and Application. *In People and computers IV: Proceedings of the Fourth Conference of the British Computer Society*, pages 35–62. Cambridge University Press.
- JONES, D. G. et ENDSLEY, M. R. (1996). Sources of Situation Awareness Errors in Aviation. *Aviation, Space, and Environmental Medicine*, 67(6):507–512.
- JOURDE, F. (2011). *Collecticiel et Multimodalité : spécification de l'interaction, la notation COMM et l'éditeur e-COMM*. Thèse de doctorat, Université J. Fourier de Grenoble.
- JOURDE, F. et LAURILLAU, Y. (2010). e-COMM, un éditeur pour spécifier l'interaction multimodale et multiutilisateur. *In Actes de la 22ème Conférence Francophone sur l'Interaction Homme-Machine (IHM '10)*, pages 225–228.
- JOURDE, F., LAURILLAU, Y. et NIGAY, L. (2010). COMM Notation for Specifying Collaborative and Multimodal Interactive Systems. *In Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '10)*, pages 125–134. ACM Press.
- KABER, D. B., WRIGHT, M. C., PRINZEL, LAWRENCE J., I. et CLAMANN, M. P. (2005). Adaptive Automation of Human-Machine System Information-Processing Functions. *Human Factors*, 47(4):730–741.

- KEKENBOSCH, C. et DENHIÈRE, G. (1988). L'activation et la diffusion de l'activation. *L'année psychologique*, 88(2):237–256.
- KIESEL, A., STEINHAUSER, M., WENDT, M., FALKENSTEIN, M., JOST, K., PHILIPP, A. M. et KOCH, I. (2010). Control and Interference in Task Switching—A Review. *Psychological Bulletin*, 136(5):849–74.
- KIRLIK, A. et STRAUSS, R. (2003). A Systems Perspective on Situation Awareness I: Conceptual Framework, Modeling, and Quantitative Measurement. Rapport technique, Naval Training Systems Center, Orlando, FL.
- KLEIN, G. A. (1993). A Recognition-Primed Decision (RPD) Model of Rapid Decision Making. In *Decision Making in Action: Models and Methods*, chapitre 6, pages 138–147.
- KLEIN, G. A. (2000). Analysis of Situation Awareness from Critical Incident Reports. In ENDSLEY, M. R. et GARLAND, D. J., éditeurs : *Situation Awareness Analysis and Measurement*, chapitre 3, pages 51–71. Lawrence Erlbaum Associates.
- KLEIN, G. A., CALDERWOOD, R. et MACGREGOR, D. (1989). Critical Decision Method for Eliciting Knowledge. *IEEE Transactions on Systems, Man and Cybernetics*, 19(3):462–472.
- KOCH, I., GADE, M., SCHUCH, S. et PHILIPP, A. M. (2010). The role of inhibition in task switching: A review. *Psychonomic Bulletin & Review*, 17(1):1–14.
- LABARTHE, J.-P., de la GARZA, C. et GRAGLIA, L. (2011). Contribution de l'ergonomie à la maîtrise des risques industriels pour la conception des moyens de conduite d'un nouveau réacteur. In *Actes du 46ème congrès international de la SELF*, pages 271–276.
- LATORELLA, K. A. (1996). *Investigating Interruptions: Implications for Flightdeck Performance*. Thèse de doctorat, State University of New York at Buffalo.
- LEAL, A., BOUCHET, J., LANGLOIS, G. et JOURDE, F. (2009). Conception de l'Interaction Homme-Machine et Partage d'Autorité : Application aux Systèmes de Drones. In *Actes de la 21ème Conférence Francophone sur l'Interaction Homme-Machine (IHM'09)*, pages 283–290. ACM Press.
- LEPLAT, J. (1997). *Regards sur l'activité en situation de travail. Contribution à la psychologie ergonomique*. Presses Universitaires de France.
- LIMBOURG, Q. et VANDERDONCKT, J. (2003). Comparing Task Models for User Interface Design. In DIAPER, D. et STANTON, N. A., éditeurs : *The Handbook of Task Analysis for Human-Computer Interaction*, chapitre 6, pages 135–154. Lawrence Erlbaum Associates.

BIBLIOGRAPHIE

- LIMBOURG, Q., VANDERDONCKT, J., MICHOTTE, B., BOUILLON, L., FLORINS, M. et TREVISAN, D. G. (2004). USIXML: A User Interface Description Language for Context-Sensitive User Interfaces. *In Proceedings of the ACM AVI'2004 Workshop "Developing User Interfaces with XML: Advances on User Interface Description Languages"*, pages 55–62.
- LOGAN, G. D. (2003). Executive Control of Thought and Action: In Search of the Wild Homunculus. *Current Directions in Psychological Science*, 12(2):45–48.
- LOGAN, G. D. et BUNDESEN, C. (2003). Clever Homunculus: Is There an Endogenous Act of Control in the Explicit Task-Cuing Procedure? *Journal of Experimental Psychology: Human Perception and Performance*, 29(3):575–599.
- LUCQUIAUD, V. (2005a). Proposition d'un noyau et d'une structure pour les modèles de tâches orientés utilisateurs. *In Actes de la 17ème Conférence Francophone sur l'Interaction Homme-Machine (IHM'05)*, pages 83–90.
- LUCQUIAUD, V. (2005b). *Sémantique et outil pour la modélisation des tâches utilisateur: N-MDA*. Thèse de doctorat, Université de Poitiers / ENSMA / INRIA.
- MACLEOD, C. M. (1991). Half a Century of Research on the Stroop Effect: An Integrative Review. *Psychological Bulletin*, 109(2):163–203.
- MARTINIE, C., PALANQUE, P., NAVARRE, D., WINCKLER, M. et POUPART, E. (2011). Model-Based Training: An Approach Supporting Operability of Critical Interactive Systems: Application to Satellite Ground Segments. *In Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS'11)*, pages 53–62.
- MAYR, U. (2009). Sticky plans: Inhibition and binding during serial-task control. *Cognitive Psychology*, 59(2):123–153.
- MAYR, U. et KEELE, S. (2000). Changing internal constraints on action: the role of backward inhibition. *Journal of Experimental Psychology: General*, 129(1):4–26.
- MAYR, U. et KLIEGL, R. (2003). Differential effects of cue changes and task changes on task-set selection costs. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 29(3):362–372.
- McFARLANE, D. C. (1997). Interruption of People in Human-Computer Interaction: A General Unifying Definition of Human Interruption and Taxonomy. Rapport technique, Navy Center for Applied Research in Artificial Intelligence Information Technology Division.
- McFARLANE, D. C. (2002). Comparison of Four Primary Methods for Coordinating the Interruption of People in Human-Computer Interaction. *Human-Computer Interaction*, 17(1):63–139.

- McFARLANE, D. C. et LATORELLA, K. A. (2002). The Scope and Importance of Human Interruption in Human-Computer Interaction Design. *Human-Computer Interaction*, 17(1):1–61.
- MCGUINNESS, B. et FOY, L. (2000). A subjective measure of Situation Awareness: the Crew Awareness Rating Scale (CARS). In *Proceedings of the 1st Human Performance, Situation Awareness and Automation Conference*, Savannah, Georgia, USA.
- MEIRAN, N. (1996). Reconfiguration of Processing Mode Prior to Task Performance. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(6):1423–1442.
- MEIRAN, N. (2000). Reconfiguration of Stimulus Task Sets and Response Task Sets during Task Switching. In MONSELL, S. et DRIVER, J., éditeurs : *Control of Cognitive Processes: Attention and Performance XVIII*, chapitre 16, pages 377–399. MIT Press.
- METH, H., MAEDCHE, A. et EINOEDER, M. (2012). Exploring Design Principles of Task Elicitation Systems for Unrestricted Natural Language Documents. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS'12)*, pages 205–210.
- MONK, C. A., TRAFTON, J. G. et BOEHM-DAVIS, D. A. (2008). The Effect of Interruption Duration and Demand on Resuming Suspended Goals. *Journal of experimental psychology: Applied*, 14(4):299–313.
- MONSELL, S. (2003). Task Switching. *Trends in Cognitive Sciences*, 7(3):134–140.
- MONSELL, S., YEUNG, N. et AZUMA, R. (2000). Reconfiguration of task-set: Is it easier to switch to the weaker task? *Psychological Research*, 63(3-4):250–264.
- MORAY, N. (2004). Où sont les neiges d'antan ? In VINCENZI, D. A., MOULOUA, M. et HANCOCK, P. A., éditeurs : *Human Performance, Situation Awareness and Automation Conference: Current Research and Trends*, pages 1–31. Keynote Speech.
- MORI, G., PATERNÒ, F. et SANTORO, C. (2002). CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. *IEEE Transactions on Software Engineering*, 28(8):797–813.
- MOUSSA, F., KOLSKI, C. et RIAHI, M. (2000). A model based approach to semi-automated user interface generation for process control interactive applications. *Interacting with Computers*, 12(3):245–279.
- NASA (2010). Human Integration Design Handbook. Rapport technique, National Aeronautics and Space Administration.

BIBLIOGRAPHIE

- NEHME, C. E., SCOTT, S. D., CUMMINGS, M. L. et YUMI FURUSHO, C. (2006). Generating Requirements for Futuristic Heterogenous Unmanned Systems. *In Proceedings of HFES 2006: 50th Annual Meeting of the Human Factors and Ergonomics Society*, San Francisco, CA.
- NEISSER, U. (1976). *Cognition and Reality: Principles and Implications of Cognitive Psychology*. W.H. Freeman.
- NOFI, A. A. (2000). Defining and Measuring Shared Situational Awareness. Rapport technique, Center for Naval Analyses.
- NOGIER, J.-F., BOUILLOT, T. et LECLERC, J. (2011). *Ergonomie des interfaces - Guide pratique pour la conception des applications web, logicielles, mobiles et tactiles*. Dunod.
- NTSB (1999). Aircraft Accident Report - Runway Overrun during Landing American Airlines Flight 1420 McDonnell Douglas MD-82, N215AA, Little Rock, Arkansas. Rapport technique, NATIONAL TRANSPORTATION SAFETY BOARD.
- OULASVIRTA, A. et SAARILUOMA, P. (2004). Long-term working memory and interrupting messages in human-computer interaction. *Behaviour & Information Technology*, 23(1):53–64.
- OULASVIRTA, A. et SAARILUOMA, P. (2006). Surviving task interruptions: Investigating the implications of long-term working memory theory. *International Journal of Human-Computer Studies*, 64(10):941–961.
- PALANQUE, P., LADRY, J.-f., BARBONI, E., NAVARRE, D. et WINCKLER, M. (2009a). Une Approche Formelle pour l'Evaluation de la Tolérance aux Interruptions des Système Interactifs. *In Actes de la 21ème Conférence Francophone sur l'Interaction Homme-Machine (IHM'09)*, pages 141–150.
- PALANQUE, P., WINCKLER, M., LADRY, J.-F., ter BEEK, M. H., FACONTI, G. et MASSINK, M. (2009b). A Formal Approach Supporting the Comparative Predictive Assessment of the Interruption-Tolerance of Interactive Systems. *In Proceedings of the 1st ACM SIGCHI symposium on Engineering Interactive Computing Systems (EICS'09)*, pages 211–220. ACM Press.
- PALANQUE, P. A. et BASTIDE, R. (1994). Petri net based design of user-driven interfaces using the interactive cooperative objects formalism. *In Proceedings of Design, Specification and Verification of Interactive Systems*, pages 383–400. Springer Verlag.
- PATERNÒ, F. (2003). ConcurTaskTrees: An Engineered Notation for Task Models. *In The Handbook of Task Analysis for Human-Computer Interaction*, chapitre 24, pages 483–500. Lawrence Erlbaum Associates.

- PATERNÒ, F., MANCINI, C. et MENICONI, S. (1997). ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. *In INTERACT '97 – Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction*, pages 362–369.
- PETERSON, J. L. (1981). *Petri Net Theory and the Modeling of Systems*. Prentice Hall.
- PEW, R. W. (2000). The State of Situation Awareness Measurement: Heading Toward the Next Century. *In ENDSLEY, M. R. et GARLAND, D. J., éditeurs : Situation Awareness Analysis and Measurement*, chapitre 2, pages 33–47. Lawrence Erlbaum Associates.
- PRESS, M. (1986). Situation Awareness: Let's et serious about the clue-bird. Manuscrit non publié.
- ROGERS, R. D. et MONSELL, S. (1995). Costs of a Predictable Switch between Simple Cognitive Tasks. *Journal of Experimental Psychology: General*, 124(2): 207–231.
- ROUSSEAU, R., TREMBLAY, S. et BRETON, R. (2004). *Defining and Modeling Situation Awareness: A Critical Review*, chapitre 1, pages 3–21. Ashgate Pub Ltd.
- SALAS, E., DICKINSON, T., CONVERSE, S. et TANNENBAUM, S. I. (1992). Toward an understanding of team performance and training. *In SWEZEY, R. et SALAS, E., éditeurs : Teams: Their Training and Performance*, pages 3–29.
- SALAS, E., PRINCE, C., BAKER, D. P. et SHRESTHA, L. (1995). Situation Awareness in Team Performance: Implications for Measurement and Training. *Human Factors*, 37(1):123–136.
- SALMON, P. M., STANTON, N. A., WALKER, G., BABER, C., JENKINS, D., MCMASTER, R. et YOUNG, M. S. (2008a). What really is going on? Review of Situation Awareness models for individuals and teams. *In HARRIS, D., éditeur : Theoretical Issues in Ergonomics Science*, volume 9 de *Lecture Notes in Computer Science*, pages 297–323. Springer Berlin / Heidelberg.
- SALMON, P. M., STANTON, N. A., WALKER, G. H. et GREEN, D. (2006). Situation Awareness Measurement: A review of applicability for C4i environments. *Applied Ergonomics*, 37(2):225–238.
- SALMON, P. M., STANTON, N. A., WALKER, G. H., JENKINS, D. P., BABER, C. et MCMASTER, R. (2008b). Representing Situation Awareness in collaborative systems: A case study in the energy distribution domain. *Ergonomics*, 51(3):367–384.

BIBLIOGRAPHIE

- SALMON, P. M., STANTON, N. A., WALKER, G. H., JENKINS, D. P., LADVA, D., RAFFERTY, L. et YOUNG, M. S. (2009). Measuring Situation Awareness in Complex Systems: Comparison of measures study. *International Journal of Industrial Ergonomics*, 39(3):490–500.
- SALMON, P. M., STANTON, N. A., WALKER, G. H., JENKINS, D. P. et RAFFERTY, L. (2010). Is it really better to share? Distributed Situation Awareness and its implications for collaborative system design. *Theoretical Issues in Ergonomics Science*, 11(1-2):58–83.
- SANER, L. D., BOLSTAD, C. A., GONZALEZ, C. et CUEVAS, H. M. (2009). Measuring and Predicting Shared Situation Awareness in Teams. *Journal of Cognitive Engineering and Decision Making*, 3(3):280–308.
- SCAPIN, D. L. et PIERRET-GOLBREICH, C. (1989). MAD : une méthode analytique de description des tâches. In *Actes du colloque sur l'ingénierie des Interfaces Homme-Machine*, Sophia-Antipolis, France.
- SCHLIENGER, C., DRAGICEVIC, P., OLLAGNON, C. et CHATTY, S. (2006). Les transitions visuelles différenciées : principes et applications. In *Actes de la 18ème Conférence Francophone sur l'Interaction Homme-Machine (IHM'06)*, pages 59–66.
- SELCON, S. J. et TAYLOR, R. M. (1990). Evaluation of the Situational Awareness Rating Technique (SART) as a tool for aircrew systems design. In *Situational Awareness in Aerospace Operations*, chapitre 5. Advisory Group for Aerospace Research and Development (AGARD).
- SHEARER, K., BUNKE, H. et VENKATESH, S. (2001). Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34(5):1075–1091.
- SHEPHERD, A. (1985). Hierarchical task analysis and training decisions. *Programmed Learning and Educational Technology*, 22(2):162–176.
- SHIVELY, R. J., BRICKNER, M. et SILBIGER, J. (1997). A Computational Model of Situational Awareness Instantiated in MIDAS. In *Proceedings of the 9th International Symposium on Aviation Psychology*, pages 1454–1459.
- SIBERTIN-BLANC, C. (1985). High-level Petri nets with data structure. In *Proceedings of 6th European Workshop on Application and Theory of Petri Nets*.
- SINNIG, D., WURDEL, M., FORBRIG, P., CHALIN, P. et KHENDEK, F. (2007). Practical Extensions for Task Models. In *Proceedings of the 6th international conference on Task models and diagrams for User Interface design (TAMODIA'07)*, numéro LNCS 4849, pages 42–55.

- SMITH, B. R. et TYLER, S. W. (1997). The design and application of MIDAS: A constructive simulation for human-system analysis. *In Second Simulation Technology and Training Conference*.
- SMITH, K. et HANCOCK, P. A. (1995). Situation Awareness Is Adaptive, Externally Directed Consciousness. *Human Factors*, 37(1):137–148.
- SPEIER, C., VESSEY, I. et VALACICH, J. S. (2003). The Effects of Interruptions, Task Complexity, and Information Presentation on Computer-Supported Decision-Making Performance. *Decision Sciences*, 34(4):771–798.
- STANTON, N. A., SALMON, P. M., WALKER, G. H. et JENKINS, D. P. (2009). Genotype and Phenotype Schemata as Models of Situation Awareness in Dynamic Command and Control Teams. *International Journal of Industrial Ergonomics*, 39(3):480–489.
- STANTON, N. A., STEWART, R., HARRIS, D., HOUGHTON, R. J., BABER, C., MCMASTER, R., SALMON, P. M., HOYLE, G., WALKER, G. H., YOUNG, M. S., LINSSELL, M., DYMOTT, R. et GREEN, D. (2006). Distributed situation awareness in dynamic systems: theoretical development and application of an ergonomics methodology. *Ergonomics*, 49(12-13):1288–1311.
- STEWART, T. R. et LUSK, C. M. (1994). Seven Components of Judgmental Forecasting Skill: Implications for Research and Improvement of Forecasts. *Journal of Forecasting*, 13:579–599.
- STRAUSS, R. et KIRLIK, A. (2003). A Systems Perspective on Situation Awareness II: Experimental Evaluation of a Modeling & Measurement Technique. Rapport technique, Naval Training Systems Center, Orlando, FL.
- STROOP, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18(6):643–662.
- TARBY, J.-C. et BARTHET, M.-F. (1996). The Diane+ Method. *In Proceedings of Computer-Aided Design of User Interfaces*, pages 95–119.
- TAYLOR, R. M. (1990). Situational Awareness Rating Technique (SART): The development of a tool for aircrew systems design. *In Situational Awareness in Aerospace Operations*, chapitre 3. Advisory Group for Aerospace Research and Development (AGARD).
- TRAFTON, J. G., ALTMANN, E. M. et BROCK, D. P. (2005). Huh, what was I doing? How people use environmental cues after an interruption. *In Proceedings of the 49th Annual Meeting Human Factors and Ergonomics Society (HFES)*, pages 468–472.

BIBLIOGRAPHIE

- TRAFTON, J. G., ALTMANN, E. M., BROCK, D. P. et MINTZ, F. E. (2003). Preparing to resume an interrupted task: effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human-Computer Studies*, 58(5):583–603.
- TRAFTON, J. G. et MONK, C. A. (2007). Task Interruptions. *Reviews of Human Factors and Ergonomics*, 3(1):111–126.
- TVERSKY, A. (1977). Features of Similarity. *Psychological Review*, 84(4):327–352.
- TYLER, S. W., NEUKOM, C., LOGAN, M. et SHIVELY, R. J. (1998). The MIDAS Human Performance Model. In *Proceedings of the 42nd Annual Meeting Human Factors and Ergonomics Society*, pages 320–324.
- van der VEER, G. C., LENTING, B. F. et BERGEVOET, B. A. J. (1996). GTA: Groupware Task Analysis - Modeling Complexity. *Acta Psychologica*, 91:297–322.
- VENEMA, D. C. (1999). *The N-UAN; a New User Action Notation*. Thèse de master, Vrije Universiteit Amsterdam.
- VERNIER, F. et NIGAY, L. (2001). A Framework for the Combination and Characterization of Output Modalities. *Interactive Systems Design, Specification, and Verification*, LCNS 1946:35–50.
- VILLAREN, T., COPPIN, G. et LEAL, A. (2012a). Modeling Task Transitions to Help Designing for Better Situation Awareness. In *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems (EICS '12)*, volume 1, pages 195–204, Copenhagen, Denmark. ACM Press.
- VILLAREN, T., LEAL, A. et COPPIN, G. (2012b). Contribution de la méthodologie SAMANTA à l'ergonomie d'interfaces en support des changements de tâche. In *Actes de la conférence Ergo'IHM 2012*, Biarritz, France. (accepté).
- VILLAREN, T., MADIER, C., LEGRAS, F., LEAL, A., KOVACS, B. et COPPIN, G. (2010). Towards a Method for Context-Dependent Allocation of Functions. In *Proceedings of the 2nd Conference on Human Operating Unmanned Systems (HUMOUS'10)*, Toulouse, France. ONERA-ISAE.
- WALKER II, J. Q. (1989). A Node-Positioning Algorithm for General Trees. Rapport technique, The University of North Carolina at Chapel Hill.
- WENDT, M. et KIESEL, A. (2008). The impact of stimulus-specific practice and task instructions on response congruency effects between tasks. *Psychological Research*, 72:425–432.
- WICKENS, C. D., MCCARLEY, J. S. et THOMAS, L. (2003). Attention-Situation Awareness (A-SA) Model. In FOYLE, D. C., GOODMAN, A. et HOOEY, B. L., éditeurs : *Proceedings of the 2003 Conference on Human Performance Modeling of Approach and Landing with Augmented Displays*, pages 189–225.

BIBLIOGRAPHIE

- ZACHARIAS, G. L., MIAO, A. X., ILLGEN, C. et YARA, J. M. (1996). SAMPLE: Situation Awareness model for pilot-in-the-loop evaluation. *In Proceedings of the 1st Annual Conference on Situation Awareness in the Tactical Air Environment.*
- ZIJLSTRA, F. R. H., ROE, R. A., LEONORA, A. B. et KREDIET, I. (1999). Temporal factors in mental work: Effects of interrupted activities. *Journal of Occupational and Organizational Psychology*, 72:163–185.
- ZSAMBOK, C. E. (1997). Naturalistic Decision Making: Where Are We Now? *In* ZSAMBOK, C. E. et KLEIN, G. A., éditeurs : *Naturalistic Decision Making*, chapitre 1, pages 3–16.



ANNEXES



Complément bibliographique : méthodes d'extraction des connaissances

Cette annexe présente dans un premier temps un aperçu des méthodes d'analyse cognitive de tâche (*Cognitive Task Analysis* ou CTA) existantes. Dans un second temps, deux méthodes sont détaillées.

- La méthode *Critical Decision Method* (CDM) de Klein *et al.* (1989), utilisée par Stanton *et al.* (2006) dans la construction des réseaux du modèle DSA (section 3.3.2 page 65), en particulier le réseau propositionnel décrivant les éléments de SA du modèle. Cette méthode est décrite dans la section A.2.
- La méthode *hybrid Cognitive Task Analysis* (analyse cognitive de tâche hybride, hCTA) de Nehme *et al.* (2006), palliant l'absence de systèmes existants, et donc d'opérateurs sur lesquels s'appuyer pour la réalisation d'une CTA, en proposant de définir des scénarios prospectifs. Cette méthode est décrite dans la section A.3.

A.1 L'ANALYSE COGNITIVE DE TÂCHE

Le terme *Cognitive Task Analysis* (CTA), « analyse cognitive de tâche » en français, regroupe l'ensemble des méthodes et outils permettant de comprendre « la façon dont les gens pensent » (Crandall *et al.*, 2006). Un grand nombre de méthodes, modèles et outils ont été proposés afin de réaliser ce but. Crandall *et al.* (2006) en répertorient plus de soixante, en s'appuyant notamment sur un état de l'art exhaustif mis en ligne par la MITRE Corporation¹. Ces méthodes sont classées par Crandall *et al.* (2006) selon quatre catégories, correspondant à quatre manières de collecter les données :

Entretiens. Ces méthodes consistent à interroger un ou plusieurs opérateurs à la fois, en suivant une structure de questions pré-définies (entretiens dirigés), en élaborant des questions à partir d'idées directrices (entretien semi-guidé) ou de manière plus libre (mais néanmoins formalisée). Ces méthodes ont l'avantage de permettre d'accéder à des informations qui auraient pu être manquées en utilisant d'autres méthodes car l'analyse peut poser des questions précises. Par exemple, si une observation ou une capture automatisée n'est pas réalisée au moment où se passe un événement, on manque cet événement alors qu'il est

1. Anciennement disponible à l'adresse <http://mentalmodels.mitre.org>, ce site n'est plus maintenu mais une version du 25/03/10 est archivée sur <http://web.archive.org>.

possible de revenir sur cet évènement et de le décortiquer en entretien. Bien que les entretiens demandent des moyens logistiques peu importants², l'accès aux opérateurs experts est souvent difficile, leur nombre et leur disponibilité étant limités.

Parmi les méthodes d'entretien existantes, nous présentons *Critical Decision Method* (CDM) dans le paragraphe A.2, utilisée par Stanton *et al.* (2006) pour la construction de leur modèle DSA notamment. Citons également la méthode *Goal-Direct Task Analysis* (GDTA), utilisée par Endsley et ses collègues afin d'extraire les buts et sous-buts des missions du système étudiés et les exigences en termes de SA associées à chacun de ces buts, représentées ensuite sous la forme de tableaux à 3 colonnes, correspondant aux 3 niveaux du modèle linéaire de la *Situation Awareness* (perception, compréhension, projection). Voir aussi Endsley *et al.* (2003) ou le paragraphe 3.2.1.1.

Auto-évaluations. Les méthodes d'auto-évaluation consistent à collecter les données des opérateurs à travers des questionnaires dont le format peut être plus ou moins structuré, voire totalement ouvert (type « journal intime »). Ces méthodes ont été décrites en partie dans le cadre de la mesure de la SA (section 3.4) mais peuvent bien sûr aborder des sujets plus larges que les exigences en termes de SA, comme par exemple les tâches en jeu. Les questionnaires sont un compromis intéressant lorsque l'accès à l'opérateur n'est pas aisé, lui laissant le temps de compléter le questionnaire lorsqu'il est disponible.

La liberté laissée à l'opérateur (questions fermées/ouvertes, ou expression libre) influence sur la précision et la qualité des réponses, qui varient d'un sujet à l'autre, produisant des résultats hétérogènes. La motivation des opérateurs lors de la complétion de ces évaluations influence également la qualité des données collectées. Afin de pallier ces inconvénients, il peut être intéressant de revenir, au cours d'un entretien, sur un questionnaire rempli par l'opérateur afin d'approfondir certains points jugés intéressants par l'intervieweur.

Observations. Ces méthodes permettent d'avoir un accès direct à l'activité de l'opérateur dans la réalisation de ses missions. Plusieurs techniques d'observations existent, s'appuyant sur des formats de collecte (et de codage) de données plus ou moins précis, définis en amont. Cette famille de méthode nécessitent de la part de l'observateur une rigueur dans la collecte d'information, notamment lorsqu'il est peu familier du système. En effet, un observateur ayant une connaissance du système saura détecter les événements et actions importants par rapport à un observateur neutre, qui pourrait rater ces faits s'il ne suit pas une grille d'observation définie.

Les observations peuvent s'avérer compliquées en termes de logistique, notamment quand elles sont accompagnées par des moyens audiovisuels particuliers (enregistrement vidéo, diffusion dans une salle voisine...).

2. un papier et un crayon peuvent suffire, même si l'usage d'un dictaphone, avec l'accord de l'interviewé, ou d'un deuxième analyste est fortement conseillé

Enfin, les méthodes d'observation peuvent également être couplées avec des entretiens auprès des opérateurs, que ce soit a posteriori afin de revenir sur certains faits marquants de la mission réalisée (parfois appuyés de rejeux lorsque ceux-ci sont disponibles), ou sous la forme de questions en cours de réalisation, demandant à l'opérateur d'expliquer à haute voix ce qu'il fait (comme pour la mesure SASHA-L par exemple, interrogeant l'opérateur sur sa SA en temps réel, voir §3.4.2.4).

Captures automatisées. Ces méthodes consistent à ajouter des points de capture sur différents aspects de la mission tout au long de la réalisation de celle-ci. Cette automatisation implique que la mission soit réalisée sur un système où l'on peut ajouter de tels points de capture. La méthode SAGAT (Endsley, 1995a) s'inscrit dans cette famille de méthode de capture, avec un focus sur la capture des éléments de SA requis pour la réalisation de certaines tâches (pour un descriptif détaillé de cette méthode, le lecteur pourra consulter le paragraphe 3.4.2.1). D'autres méthodes similaires peuvent être imaginées, demandant à l'opérateur en cours de mission de justifier ses choix ou de décrire d'autres aspects cognitifs (autre que la SA) de sa mission.

Ainsi, ces différents types de méthodes constituent une grande boîte à outils pour la personne en charge de l'extraction des connaissances. La sélection d'une ou plusieurs méthodes de collecte de données dépend de l'objectif poursuivi par l'analyste.

A.2 CRITICAL DECISION METHOD (CDM)

Décrite par Crandall *et al.* (2006, chapitre 5) comme une des « méthodes les plus puissantes d'extraction de connaissance », la méthode des « décisions critiques » (*Critical Decision Method* ou CDM) (Klein *et al.*, 1989) consiste à interroger des opérateurs ayant de l'expérience dans l'utilisation du système considéré au sujet d'incidents survenus lors de la réalisation de missions passées. Cette méthode, classée dans la catégorie des méthodes d'analyse cognitive de tâche basée sur les incidents est appliquée au cours d'entretiens avec les opérateurs. La version initiale de cette méthode se décompose en cinq étapes (figure A.1) :



FIGURE A.1 : les cinq étapes de la méthode *Critical Decision Method* (Klein *et al.*, 1989)

- 1. Sélection de l'incident :** la première étape consiste à passer en revue avec l'opérateur les missions au cours desquelles des événements ou incidents importants sont survenus et de faire une liste de ces incidents. Les événements

considérés doivent être non routiniers et faire appel à l'expertise de l'opérateur pour la résolution du problème considéré. Généralement, un incidents plus saillant que les autres apparait rapidement dans la discussion et est donc sélectionné pour mener à bien l'entretien. Klein *et al.* (1989) conseillent d'éviter les événements dont la saillance émane du décès de personnes impliquées ou ceux dans lesquels l'opérateur ne jouait pas de rôle « clé » en termes décisionnels.

2. **Restitution non structurée** : lorsque l'incident a été sélectionné et validé, l'opérateur le décrit du début à la fin, sans être interrompu par la personne réalisant l'entretien, excepté pour demander de clarifier certains points, le cas échéant. Cette étape permet de remplir trois objectifs :
 - Pour la personne réalisant l'entretien, cela crée un « contexte » permettant de comprendre comme l'incident a été vécu par l'opérateur et permettre d'orienter les questions en fonction de cette perception ;
 - Pour l'opérateur interviewé, ce souvenir permet de ré-activer en mémoire les différentes phases de l'incident et de préparer les questions à venir ;
 - Enfin, cette phase d'écoute permet d'instaurer une relation de confiance entre les deux interlocuteurs.
3. **Construction de la chronologie** : à partir de la restitution complète de l'incident, l'intervieweur construit une chronologie des différents événements de l'incident, en précisant leur enchaînement temporel et la durée de chacune d'entre elles. Les événements considérés sont à la fois des événements que l'on peut vérifier objectivement (une alerte sonore, l'arrivée d'un message, l'entrée dans une zone à risque...) et des événements plus subjectifs, portant sur la perception de l'opérateur et sa compréhension de la situation (par exemple, « la couleur de la fumée indiquait la présence de substances toxiques. », Klein *et al.* 1989). La construction de cette chronologie permet de corriger les contradictions temporelles détectées lors de l'étape précédente et d'extraire les faits marquants pour l'étape suivante.
4. **Identification des points de décision** : cette étape permet de définir quels sont les événements, parmi ceux identifiés dans l'étape précédente, qui ont nécessité une décision de la part de l'opérateur. Ce sont ces points de décision qui seront approfondis lors de l'entretien. Leur identification peut être simple : par exemple, si l'opérateur indique lors de sa restitution qu'il a dû prendre une décision. Mais parfois, ces jalons ne sont pas identifiés directement. Pour ce faire, l'intervieweur doit demander à l'opérateur si d'autres solutions auraient été possibles et à quels moments elles auraient dû être prises. Ces moments correspondent aux jalons recherchés.
5. **Approfondissement des points de décision** : les quatre étapes précédentes servent à préparer le terrain pour l'accès aux informations précises sur la prise de décision. C'est cette dernière étape qui permet d'approfondir chacun des points de décision identifiés précédemment et de les approfondir à travers un ensemble de questions. Ces questions sont synthétisées dans le tableau A.1.

Le but de cette étape est de comprendre le déroulement de chaque prise de décision, notamment les buts et tâches en jeu ainsi que les connaissances nécessaires à la réalisation de ces buts.

Il est notamment important d'approfondir les différents buts trop « abstraits » en buts plus spécifiques, permettant de comprendre la décomposition des tâches. De la même façon, les auteurs indiquent qu'il est parfois plus facile de comprendre les raisons de la prise d'une décision en considérant les options écartées par l'opérateur.

Notons que nous avons présenté dans ce paragraphe la méthode CDM telle que décrite initialement par Klein *et al.* (1989). Crandall *et al.* (2006) reprennent ce déroulement en groupant différentes étapes au sein de « passages », afin de limiter le temps de discussion avec les opérateurs. Cette méthode est bien sûr adaptée lors des entretiens en fonction des personnes interviewées, du temps imparti et du projet dans le cadre duquel l'entretien est mené.

La méthode CDM n'est pas applicable dans tous les cas. Dans certains domaines, il n'existe pas de vrais experts. Crandall *et al.* (2006) donnent l'exemple des services de sécurité des aéroports où le taux de renouvellement des emplois est important, ne laissant pas le temps de construire l'expérience nécessaire pour pouvoir revenir sur les incidents critiques. Dans certains cas également, ces incidents critiques n'existent pas, comme les études de consommation ou des pratiques des utilisateurs. Enfin, pour des domaines très pointus ou novateurs, l'accès à des experts s'avère difficile, car ils sont peu nombreux, voire inexistant dans le cas de systèmes totalement nouveaux.

A

A.3 ANALYSE COGNITIVE DE TÂCHE HYBRIDE

Afin de pallier l'absence de données, d'experts du domaine et d'opérateurs, et donc d'implémentation antérieures sur lesquelles reposer l'analyse cognitive de tâche, Nehme *et al.* (2006) introduisent une version hybride d'analyse cognitive de tâche (*hybrid Cognitive Task Analysis* ou hCTA) permettant, à partir de scénarios projectifs décrivant le système (une console de supervision de plusieurs drones aériens dans l'étude présentée), de définir les tâches et sous-tâches ainsi que les exigences en termes de SA associées à chacune de ces tâches.

Le processus est décomposé en quatre étapes :

- La **génération d'un scénario** décrivant la mission à travers différentes phases. L'analyste produit alors un ensemble de tâches et sous-tâches comme dans le cas d'une analyse hiérarchique des tâches (HTA, voir §2.1.2.1). Afin d'effectuer cette décomposition, l'analyste doit parfois faire des hypothèses qui doivent être explicitées ;
- La **création d'un diagramme temporel** décrivant l'enchaînement temporel des différentes tâches précédemment décrites ;
- La **génération d'exigences SA** associées aux tâches et sous-tâches, impactées par les contraintes temporelles définies lors de la deuxième étape. Le

TABEAU A.1 : questions posées lors de l'entretien en utilisant la méthode des décisions critiques (traduit de Klein *et al.*, 1989).

Type de questions	Détail des questions
Indices / Perception	« Que voyiez-vous, entendiez-vous, sentiez-vous ? »
Connaissances	« Quelles informations avez-vous utilisées pour prendre cette décision, et comment ont-elles été obtenues ? »
Analogies	« Vous êtes-vous rappelé d'une expérience passée similaire ? »
Buts	« Quels étaient vos buts spécifiques à cet instant ? »
Options	« Quelles autres options avaient-vous considérées, ou étaient disponibles ? »
Justification	« Comment avez-vous choisi cette option particulière / rejeté les autres options ? Quelle règle était suivie ? »
Expérience	« Quelle formation particulière ou expérience était nécessaire ou utile à cette prise de décision ? »
Amélioration possible	« Si la décision n'était pas la meilleure, quelle formation, connaissance ou information auraient pu vous aider ? »
Pression temporelle	« Quelle pression, en termes de temps, subissiez-vous lors de cette prise de décision ? »
Évaluation de la situation	« Imaginez qu'à cet instant, une personne venant vous remplacer vous demande de décrire la situation, comment la synthétiseriez-vous ? »
Hypothèses	« Si un élément clé de la situation avait été différent, comment votre décision aurait-elle été impactée ? »

modèle linéaire de Endsley (1995b) est utilisé pour définir, pour chaque tâche, les éléments requis sur ses trois niveaux de SA.

- La **génération d'une échelle de décision** décrivant, pour chaque décision, quelles sont les prérogatives du système et celles de l'opérateur ainsi que les informations que chacun doit avoir à sa disposition pour prendre la décision.

Deux variantes de cette échelle doivent ensuite être construites, l'une décrivant les informations à afficher pour chaque niveau de connaissance, l'autre pour décrire les différents niveaux d'autonomie de ce système.

Cette dernière étape conduit à la création d'interfaces appuyant les différentes échelles construites; ces interfaces seront ensuite évaluées vis-à-vis des exigences formulées en termes de SA et de prise de décision.

Le processus hCTA permet de décrire à la fois les tâches et les exigences en termes d'informations pour la prise de décision et la réalisation d'un scénario sur un futur système. C'est un processus entier, dont la sortie est un ensemble d'interfaces répondant aux différentes descriptions construites. Néanmoins, la construction de ces quatre descriptions (décomposition hiérarchiques, temporelles, exigences en termes de SA et exigences pour la prise de décision) nous semble parfois redondante et pourraient être simplifiées.

Nous remarquons ainsi que, parmi les quatre étapes, les deux premières concernent la description des tâches et sous-tâches de manière hiérarchique et temporelle, qui pourrait être synthétisée au travers de la création d'un arbre de tâche (en utilisant la notation CTT par exemple).

Enfin, dans le but de diffuser cette méthodologie et de « convaincre l'industrie que des bénéfices peuvent être tirés de la valorisation de ce type de méthode » (Cummings *et al.*, 2012b), une version gratuite d'un outil soutenant la méthode hCTA est mise à disposition sur le site du laboratoire *Humans & Automation Lab* : <http://web.mit.edu/aeroastro/labs/halab/inventions.shtml>.



Description fonctionnelle de l'outil SAMANTA

Dans le cadre de la conception de l'outil soutenant de la méthodologie SAMANTA (chapitre 6), nous avons réalisé une analyse fonctionnelle préalable afin de définir les fonctions que devait prendre en charge l'outil, afin d'appuyer la réalisation des étapes 3 et 4.

Nous définissons donc deux *fonctions principales* (FP) correspondant à ces deux étapes. Ces fonctions principales sont décomposées en *fonctions secondaires* (FS), décrites précisément dans la suite de cette annexe. Pour chaque fonction secondaire, seront notamment définies des contraintes qui limitent la liberté dans le développement futur de l'outil, dans le but de répondre précisément au besoin formulé pour chaque fonction principale.

B.1 FP 1 : « DÉFINIR LES CONTEXTES ASSOCIÉS À CHAQUE TÂCHES DE L'ARBRE »

Cette première fonction principale traduit le besoin correspondant à l'étape 3 de la méthodologie SAMANTA (section 5.2.4), où l'analyste doit définir, pour chaque tâche de l'arbre CTT, quelles sont les exigences en termes de *Situation Awareness* pour l'opérateur.

Les fonctions secondaires (FS) de cette fonction principale englobent les fonctions de chargement et d'affichage des deux modèles (construits auparavant), de navigation dans ces modèles et de modification de certaines informations relatives aux tâches et éléments de situation.

B.1.1 FS 1.1 : « Charger, afficher et naviguer dans l'arbre des tâches »

B.1.1.1 Description

Cette fonction correspond au chargement du fichier décrivant l'arbre des tâches, et à l'affichage sur l'interface de cet arbre afin de permettre à l'analyste de le visualiser et d'y naviguer. Les options de navigation requises sont « classiques » : déplacement horizontal et vertical (fonctions regroupées sous le terme de « *pan* » en anglais) et agrandissement / rétrécissement (*zoom*) de la vue.



B.1.1.2 Contrainte

L'outil doit être compatible avec le format de fichier de sortie de l'éditeur d'arbre de tâche CTTE (section 6.3.2.1), il doit pouvoir interpréter ce fichier afin d'en extraire les informations nécessaires à l'affichage de l'arbre sur l'interface.

B.1.2 FS 1.2 : « Charger, afficher et naviguer dans le réseau propositionnel des éléments de situation »

B.1.2.1 Description

Cette fonction correspond au chargement du fichier décrivant le réseau propositionnel des éléments de situation, et à l'affichage sur l'interface de ce réseau afin de permettre à l'analyste de le visualiser et d'y naviguer. Les options de navigation définies sont les mêmes que pour la navigation dans l'arbre des tâches, à savoir le déplacement horizontal/vertical et l'agrandissement/rétrécissement de la vue.

B.1.2.2 Contraintes

B

- L'outil doit être compatible avec le format de fichier de sortie de l'éditeur de graphe *yEd Graph Editor* (section 6.3.2.2), il doit pouvoir interpréter ce fichier afin d'en extraire les informations nécessaires à l'affichage du graphe sur l'interface.
- A la différence des arbres pour lesquels l'algorithme d'agencement visuel des tâches reste relativement simple, la structure visuelle d'un arbre suivant des règles précises, l'affichage d'un graphe peut s'avérer compliqué, en fonction du nombre de nœuds et d'arcs, notamment si l'on veut optimiser cet affichage pour éviter tout effet de « désordre » (*clutter*).

De nombreux algorithmes d'agencement des éléments et arêtes d'un graphe existent, mais la définition ou l'optimisation de ce type d'affichage n'étant pas le sujet de cette thèse, nous ne nous focalisons pas sur l'implémentation de tels algorithmes, souvent complexes.

Afin d'anticiper les éventuels soucis d'affichage, il est requis que la définition de l'agencement des nœuds soit effectuée en amont et décrite en tant que donnée d'entrée du graphe : le fichier décrivant les nœuds et arêtes devra intégrer des informations sur la position des nœuds et libellés des arcs.

B.1.3 FS 1.3 : « Afficher et modifier les informations relatives à une tâche de l'arbre »

B.1.3.1 Description

Il est important que l'utilisateur-ergonome puisse accéder rapidement à l'ensemble des informations relatives aux tâches de l'arbre CTT, à savoir :

- Nom et description de la tâche ;
- Catégorie CTT de la tâche (abstraite, utilisateur, système, interactive) ;
- Son caractère itératif et/ou optionnel ;
- Son caractère factice ;
- L'ensemble des éléments de situation définissant le contexte de la tâche, s'il est défini.

Certains de ces éléments doivent pouvoir être modifiés :

- La description de la tâche ;
- Le caractère factice ou non de la tâche ;
- Le contexte associé à la tâche. Cette modification est l'objet de la fonction secondaire 1.3 (§ B.1.5)

B.1.3.2 Contraintes

L'arbre de tâches doit avoir été chargé et affiché afin de pouvoir accéder aux différentes tâches. De plus, l'affichage et la modification des contextes associés à chaque tâche ne peut être effectué que si le graphe de situation a également été chargé.

B

B.1.4 FS 1.4 : « Afficher et modifier les informations relatives à un élément de situation »

B.1.4.1 Description

De la même manière que la fonction précédent, il s'agit pour cette fonction de permettre à l'utilisateur-ergonome d'afficher les informations relatives à un élément de situation, à savoir :

- Le nom de l'élément ;
- Les éléments auxquels il est directement liés dans le graphe ;
- Les tâches de l'arbre auxquelles il est associé ;
- Les tâches de l'arbre pour lesquelles il est considéré comme saillant (§ 5.3.4) ;
- Les métadonnées (sous forme de tags) définies pour cet élément.

Certains de ces éléments doivent pouvoir être modifiés :

- Les tâches auxquelles l'élément est associé ;
- Le caractère saillant de l'élément pour chacune des tâches auxquelles il est associé ;
- Les tags définis pour l'élément.

B.1.4.2 Contraintes

- Le graphe propositionnel doit avoir été chargé et affiché afin de pouvoir accéder aux différents éléments de situation.
- Les informations relatives aux contextes auxquels appartient un élément particulier ne peuvent être affichées qu'à la condition préalable d'avoir également chargé et affiché l'arbre des tâches.

B.1.5 FS 1.5 : « Définir les éléments de situations associés à chaque tâche »

B.1.5.1 Description

Cette fonction secondaire reformule le titre de la fonction principale dont elle est issue. En effet, il s'agit ici de permettre à l'analyste d'associer à une tâche de l'arbre l'ensemble des éléments de situation qu'il aura identifié comme étant importants pour la réalisation de cette tâche.

Ce processus d'association peut se faire de deux façons :

- Pour une tâche donnée, définie comme « tâche courante » pour l'association, en délimitant le contexte associé par la sélection de multiples éléments de situation dans le graphe ;
- Pour un élément de situation donné, en le définissant comme élément pertinent pour une ou plusieurs tâches de l'arbre.

Afin de mener à bien ce processus d'association, l'analyste doit pouvoir alterner entre l'affichage de l'arbre et celui du graphe de manière simple et fluide, afin d'assurer un accès rapide aux deux modèles.

B.1.5.2 Contraintes

L'arbre de tâches et le graphe de situation doivent avoir été chargés et affichés dans l'outil, puisqu'en effet ce processus d'association ne peut être mené qu'en présence des deux modèles ;

B.2 FP 2 : « ÉVALUER L'IMPACT DES TRANSITIONS SUR LES ÉLÉMENTS DE SITUATION »

Cette deuxième fonction principale est associée à l'étape 4 de la méthodologie SAMANTA, décrite dans la section 5.3 du chapitre précédents. Ainsi, elle prend en charge l'analyse des transitions de l'arbre, de manière individuelle, à travers l'usage de différentes méthodes de comparaison des tâches définies dans la section 5.3.

Les fonctions secondaires définies mettent en œuvre cette analyse à travers la définition des transitions à analyser, l'exécution des algorithmes associés aux méthodes proposées et l'affichage des résultats pour interprétation de l'analyste.

B.2.1 FS 2.1 : « Définir une transition à analyser »

B.2.1.1 Description

Afin de mener à bien l'analyse, l'utilisateur-ergonome doit dans un premier temps définir la transition à analyser. Comme nous l'avons présenté dans la section 5.3.1, deux méthodes lui permettent de spécifier cette transition :

- Soit la transition est directement accessible dans l'arbre des tâches, car elle est visualisée par un identificateur CTT liant deux tâches de l'arbre. Dans ce cas, l'utilisateur-ergonome doit pouvoir accéder à la transition à travers l'arbre ;
- Soit la transition n'est pas directement visible dans l'arbre, car les deux tâches qui la constituent ne sont pas directement liées (pas de parent direct commun ou alors, pas d'identificateur CTT les liant directement). Dans ce cas, il faut permettre à l'analyste de définir la transition à analyser en sélectionnant directement les deux tâches concernées.

B.2.1.2 Contraintes

L'analyse d'une transition n'étant pertinente que si des contextes ont été associés aux tâches la constituant, la contrainte imposée est le chargement et l'affichage préalable des deux modèles.

La définition des contextes associés aux tâches de la transition reste optionnelle, lors de l'analyse, l'outil pourra appliquer les règles d'héritage définies dans la section 5.2.4.3. Notons que ces règles ne peuvent être appliquées que si les tâches « feuilles » descendantes des deux tâches analysées ont des contextes associés (par définition de l'analyste ou par héritage).

B.2.2 FS 2.2 : « Analyser l'impact d'une transition sur les contextes associés »

B.2.2.1 Description

Cœur de la méthodologie SAMANTA, l'analyse des transitions est mise en œuvre à travers cette fonction secondaire. Il s'agit, pour une transition définie, de comparer les tâches constituant la transition à travers l'application des différentes méthodes décrites dans la section 5.3 :

- Affichage de statistiques et informations générales sur la transition (§ 5.3.2) ;
- Extraction des plus grands sous-graphes communs aux deux contextes de la transition (§ 5.3.3) ;
- Analyse des éléments saillants, selon les deux définitions proposées, des contextes associés à la transition (§ 5.3.4) ;
- Une mise en avant des métadonnées (tags) propres aux différents éléments de situation de la transition.

Les résultats mis à disposition de l'analyste à travers cette fonction doivent permettre de répondre à la question « comment la transition impacte les exigences en termes de *Situation Awareness* des tâches la constituant ? », ceci à travers l'affichage des informations décrites ci-dessus.

Dans le cas de l'analyse d'une transition, l'outil prendra également en compte l'intégration (ou non) des tâches parallèles aux deux tâches impliquées dans la transition (voir § 5.3.6).

B.2.2.2 Contraintes

Les différentes méthodes d'analyse introduites s'appuient sur l'analyse des contextes associés aux tâches de la transition sélectionnée. Il est ainsi nécessaire que ces contextes soient définis, soit directement par l'analyste à travers l'étape d'association pour ces tâches, ou alors en appliquant les règles d'héritages définies dans la section 5.2.4.3.

B.3 FONCTIONS COMPLÉMENTAIRES

Les fonctions suivantes ne sont pas directement nécessaires pour la mise en œuvre de la méthodologie mais elles peuvent améliorer l'expérience d'utilisation de l'outil. Nous avons défini deux fonctions complémentaires, implémentées lors du développement initial de l'outil.

B.3.1 FC 1 : « Gérer / enregistrer le projet »

Afin de permettre à l'utilisateur-ergonome de conserver une trace de ses travaux, une fonctionnalité de gestion de projet – pouvant se limiter à l'enregistrement/chargement de travaux existants – nous semble utile. Outre le nommage du projet en cours, cette fonctionnalité doit permettre d'enregistrer les modifications apportées au projet (en termes d'associations tâches / contextes) et de le retrouver plus tard afin de poursuivre l'analyse là où elle aurait été interrompue.

B.3.2 FC 2 : « Exporter le graphe sous forme d'image »

Nous proposons également d'implémenter une fonction permettant à l'analyste d'exporter le graphe de situation sous forme visuelle afin d'avoir un ensemble de fichiers images décrivant les contextes définis pour chaque tâche. L'analyste pourra notamment s'appuyer sur ces images pour la validation des associations avec des opérateurs, en les imprimant et les distribuant lors de groupes de travail par exemple.



Détails techniques sur l'outil SAMANTA

C.1 BIBLIOTHÈQUES UTILISÉES

Dans cette section, nous présentons succinctement les bibliothèques utilisées pour la construction de l'application, en précisant notamment la version utilisée, les licences sous lesquelles sont disponibles ces bibliothèques ainsi qu'une référence vers le site de la bibliothèque.

C.1.1 La bibliothèque jQuery et ses extensions

Afin de construire l'application, nous nous basons principalement sur la bibliothèque *jQuery*, disponible à l'adresse <http://jquery.com/>. Cette bibliothèque facilite notamment la manipulation des documents HTML en permettant d'accéder aux différents objets du DOM (*Document Object Model*, la structure globale d'une page HTML regroupant les composantes de cette page telles que les cadres DIV, les composants de formulaire, graphiques et textuels). La bibliothèque jQuery prend également en charge la gestion des événements, résultats de l'interaction avec l'interface, ainsi que la gestion des fichiers XML (à travers l'implémentation de méthodes génériques AJAX, *Asynchronous JavaScript And XML*).

jQuery possède l'avantage d'ajouter une couche d'abstraction dans le code afin de permettre aux scripts développés de s'adapter aux différents navigateurs utilisés. En effet, chaque grande famille de navigateur – Microsoft Internet Explorer, Mozilla Firefox, Apple Safari et Google Chrome étant les quatre acteurs majoritaires du marché des navigateurs – traitent différemment certains aspects de la manipulation de la structure DOM. Ces différences historiques tendent à s'harmoniser afin de respecter les standards définis par le W3C¹, mais elles existent encore. La bibliothèque jQuery permet de s'abstraire de la gestion de ces différences, facilitant le travail des développeurs.

La bibliothèque jQuery est distribuée sous la double licence GPL et MIT. Nous choisissons la licence MIT, donnant le droit de l'utiliser, la copier, la modifier, la

1. *World Wide Web Consortium*, <http://www.w3.org/>, l'organisme en charge des standards du Web.

fusionner, la publier, la distribuer, la vendre et d'en changer sa licence. La seule obligation est de conserver les noms des auteurs dans une notice de copyright fournie avec le fichier².

De nombreuses extensions à la bibliothèque jQuery sont disponibles sur le Web. Nous présentons dans la suite de cette section la liste des extensions que nous utilisons dans le cadre du développement de l'outil soutenant la méthodologie SAMANTA en précisant à chaque fois sous quelle licence nous l'utilisons.

Mousewheel : cette extension³ ajoute une couche d'abstraction dans le code afin de gérer de manière homogène pour le développeur les données issues de l'utilisation de la molette de la souris, quelque soit le navigateur de l'utilisateur. Elle est utilisée sous licence MIT.

Jeditable : cette extension, disponible à l'adresse <http://www.appelsiini.net/projects/jeditable>, permet d'ajouter un comportement interactif d'édition à n'importe quel élément textuel de la page Web. En ajoutant une classe CSS particulière ou à travers une commande spécifique JavaScript, cette extension permet ainsi de définir le comportement d'une zone de texte afin qu'elle devienne modifiable lorsque l'utilisateur clique (ou double-clique) dessus. Cette extension est utilisée sous licence MIT.

Tags Input : cette extension, disponible à l'adresse <http://xoxco.com/clickable/jquery-tags-input>, permet d'ajouter une fonctionnalité de gestion des tags à une zone de saisie d'un formulaire, afin de permettre à l'utilisateur de saisir une liste de tags de manière intuitive. Elle est utilisée sous licence MIT.

jQuery UI : cette extension, disponible à l'adresse <http://jqueryui.com/>, met à disposition des composants logiciels supplémentaires (*widgets*) permettant d'améliorer l'interface de l'application (gestion des onglets, des menus, etc.) Elle est utilisée sous licence MIT.

Selectmenu : cette bibliothèque, disponible à l'adresse <https://github.com/fnagel/jquery-ui/wiki/Selectmenu>, est une extension à la bibliothèque jQuery UI. Elle ajoute un composant logiciel de « menu déroulant » compatible avec la bibliothèque jQuery et les thèmes graphiques disponibles dans jQuery UI. Elle est utilisée sous licence MIT.

Rcarousel : cette extension pour jQuery UI, disponible à l'adresse <http://ryrych.github.com/rcarousel/>, permet d'ajouter un effet de « carrousel » à une liste d'objets. Cet effet permet de naviguer dans la liste d'objet à la manière d'un diaporama, chaque objet venant prendre la place du précédent lorsque l'utilisateur interagit avec le carrousel. Elle est utilisée sous licence MIT.

Touch Punch : cette extension pour jQuery UI, disponible à l'adresse <http://touchpunch.furf.com/>, permet de rendre compatibles les différents Widgets

2. <http://opensource.org/licenses/mit-license.php>

3. <https://github.com/brandonaaron/jquery-mousewheel/>

créés avec la bibliothèque jQuery UI sur des supports utilisant la technologie d'interaction tactile. Elle est utilisée sous licence MIT.

C.1.2 Construction des modèles de tâche et de situation

Au sein de l'outil développé, nous utilisons une bibliothèque particulière permettant de gérer la construction et l'affichage de l'arbre des tâches et du graphe de situation. Cette bibliothèque, appelée RaphaelJS, est disponible sous licence MIT à l'adresse <http://raphaeljs.com/>. Elle permet de créer des graphiques au format SVG ou VML (selon le navigateur cible, détecté automatiquement) et d'ajouter les éléments créés (nœuds, tâches, arcs, etc.) à la structure DOM de la page. L'utilisateur peut alors interagir directement avec ces éléments à travers les événements JavaScript correspondant.

Il s'agit d'une bibliothèque purement d'affichage et d'interaction. La gestion de la construction de l'arbre et du graphe, en particulier le positionnement de chaque élément, est laissée à la charge du développeur.

C.1.2.1 Algorithme spécifique à la construction de l'arbre des tâches

Afin de gérer le positionnement des nœuds représentant les tâches de l'arbre, nous avons adapté la bibliothèque ECOTree. Nous en avons repris l'algorithme de positionnement, issu des travaux de Walker II (1989). Cette bibliothèque, disponible sous la licence *Code Project Open License* (CPOLE)⁴ (le code source peut être librement modifié, utilisé et distribué, sans garantie de responsabilité de la part de l'auteur initial).

Cette bibliothèque a été initialement mise en place pour dessiner des arbres en utilisant une combinaison de l'élément *<Canvas>* de HTML5 pour le dessin des nœuds et des éléments DIV affichés en surimpression afin de gérer l'interaction. Nous l'avons adaptée afin de dessiner des arbres en utilisant le même algorithme et les mêmes capacités de paramétrisation (taille des nœuds, distance entre nœuds, ...) mais à travers la bibliothèque RaphaelJS afin de produire un graphique au format SVG (ou VML pour Internet Explorer) où les éléments affichés à l'écran sont interactifs directement.

C.1.2.2 Note sur la construction du graphe de situation

La construction du graphe de situation se fait directement à partir de l'extraction des données de positionnement fournies dans le fichier XML importé. Ces données sont donc issues du positionnement effectué par le logiciel *yEd Graph Editor* et l'utilisateur ayant créé le graphe. La seule modification effectuée est une adaptation du système de coordonnées de *yEd Graph Editor* à celui de RaphaelJS. Il s'agit

4. <http://www.codeproject.com/Articles/16192/Graphic-JavaScript-Tree-with-Layout>

d'une translation du référentiel de l'un vers celui de l'autre afin de définir le point en haut à gauche de la zone d'affichage comme l'origine du repère.

C.1.3 Autres bibliothèques utilisées

Dans le cadre de nos développements, nous nous sommes également appuyés sur les bibliothèques suivantes :

JSON : cette bibliothèque, disponible à l'adresse <http://www.json.org/js.html>, permet de convertir des objets JSON en chaînes de caractères et inversement. Nous utilisons ces objets pour la sauvegarde et l'exportation des projets créés par l'utilisateur (voir aussi la section C.4.2 dans cette annexe). Cette bibliothèque fait partie du domaine public, elle est donc librement utilisable.

Canvg : cette bibliothèque, disponible à l'adresse <http://code.google.com/p/canvg/>, permet de convertir une image SVG au format « Canvas », propre à JavaScript et qui pourra ensuite être exporté sous un format d'image plus répandu tel que le PNG ou le JPG. Nous utilisons cette bibliothèque pour permettre de convertir l'arbre ou le graphe de situation affichés sous forme d'image pour que l'utilisateur puisse les intégrer à d'autres travaux. Elle s'appuie sur la bibliothèque libre `RGBColor`⁵.

C.2 NAVIGATEURS COMPATIBLES

Certaines technologies utilisées dans le cadre du développement de l'outil soutenant la méthodologie SAMANTA sont issues du langage HTML5⁶ dont certaines fonctionnalités ne sont pas disponibles dans les navigateurs les plus anciens. Les navigateurs considérés sont :

- Pour les plateformes de type « ordinateurs »
 - Microsoft Internet Explorer (IE) en version 7 ou supérieure ;
 - Mozilla Firefox (FF) en version 3.6 ou supérieure ;
 - Google Chrome en version 14 ou supérieure ;
 - Apple Safari en version 5.1, navigateur fourni par défaut sous Mac OS X.
- Pour les supports mobiles :
 - Apple Safari iOS en version 3.2 ou supérieure, navigateur fourni par défaut sur les iPad ;
 - Android Browser en version 2.1 ou supérieure, navigateur fourni par défaut sur les tablettes Android ;
 - Chrome pour iPad et Android en version 18 ou supérieure ;
 - Firefox (FF) pour Android en version 14 ou supérieure.

Les principales technologies utilisées limitant la compatibilité entre les navigateurs sont les suivantes :

5. <http://www.phpied.com/rgb-color-parser-in-javascript/>

6. <http://www.w3.org/TR/html5/>

SVG/VML utilisées pour la description des graphiques. Ces technologies sont incompatibles avec les versions du navigateur intégré à Android en version inférieure ou égale à 2.3.

Technologie Web Storage utilisée pour la sauvegarde locale des données (API JavaScript *localStorage*). Ces technologies sont incompatibles avec les versions d'Internet Explorer inférieures ou égales à 7.0.

Technologie offline utilisée pour le stockage de l'application sur la plateforme (pour le fonctionnement hors connexion). Ces technologies sont incompatibles avec les version d'Internet Explorer inférieures ou égales à 9.0.

File API utilisée pour l'importation de fichier. Cette technologie est incompatible avec toutes les version d'Internet Explorer distribuées à ce jour (versions inférieures ou égales à 10), ainsi qu'avec les navigateurs mobiles « par défaut » (Safari iOS et Android Browser). Afin de pallier ces problèmes, la fenêtre modale d'importation des fichiers propose un champ de texte où l'utilisateur peut copier/coller le contenu d'un fichier XML, GraphML ou JSON ainsi qu'un deuxième champ de saisie où il peut entrer l'adresse d'un tel fichier pour l'importation. Notons que sur Android, l'utilisateur peut installer des navigateurs alternatifs Chrome ou Firefox. Sur iOS, le navigateur Chrome de Google est également disponible.

Le tableau C.1 présente la compatibilité de l'application avec les différents navigateurs. Nous avons regroupé les navigateurs en fonction des systèmes d'exploitation cible et au sein desquels ils ont été testé mais, en théorie, les versions équivalentes de ces navigateurs pour d'autres systèmes (Mozilla Firefox sous Mac OS X par exemple) devraient être fonctionnelles.

Notons que l'utilisation de l'outil avec le navigateur Internet Explorer (IE) n'est possible qu'à partir de la version 10, et dans ce cas, l'utilisateur devra importer les fichiers en copiant/collant leur contenu dans la zone de texte, c'est donc un fonctionnement en mode dégradé.

C.3 DESCRIPTION DU FORMAT DES DONNÉES D'ENTRÉE

Lors de la création d'un nouveau projet dans l'outil SAMANTA, l'utilisateur doit importer deux fichiers décrivant le modèle de tâches, issus de l'outil CTTE et le modèle de situation, sous la forme d'un graphe issus de l'outil *yEd Graph Editor*. Les formats retenus afin de décrire ces fichiers d'entrée sont deux formats similaires :

- le fichier de sortie de l'application CTTE est un fichier au format XML (*eXtended Markup Language*) ;
- le fichier de sortie de l'application *yEd Graph Editor* est un fichier au format GraphML, qui est une extension du format XML et suit donc la même logique. Lors du traitement de cette donnée d'entrée au sein de l'outil SAMANTA, ce deuxième fichier sera considéré comme un fichier XML.

TABLEAU C.1 : table de compatibilité des navigateurs pour l'outil SAMANTA.

Système	Navigateur	Version minimale
Microsoft Windows	Internet Explorer	10
	Mozilla Firefox	3.6
	Google Chrome	14
Mac OS X	Apple Safari	5.1
iPad	Apple Safari iOS	3.2
	Chrome pour iPad	18
Android	Android Browser	3.0
	Mozilla Firefox	18
	Google Chrome	14

Chaque fichier XML consiste en un ensemble hiérarchique de balises (représentées par des chevrons < >) décrivant les données du modèle. Chaque structure de données étant différente, le contenu de ces fichiers XML sont décrits par des schémas, permettant de vérifier la validité d'un fichier XML vis-à-vis d'une structure de données précise.

Plusieurs langages de description des schémas existent. Les fichiers issus de CTTE et *yEd Graph Editor* sont tous deux décrits par des schémas au format DTD (*Document Type Definition*) décrivant la structure logique et physique des documents XML auxquels ils sont rattachés.

Dans les deux sous-sections suivantes, nous décrivons succinctement la structure des fichiers utilisés en entrée, en nous focalisant sur les données nécessaires pour la construction des modèles au sein de l'outil. La référence vers le fichier DTD complet sera fournie afin de permettre au lecteur de prendre connaissance de la structure globale des fichiers décrits.

C.3.1 Description des arbres au format XML

Les fichiers XML décrivant les arbres de tâches de l'application CTTE sont formalisés par un schéma DTD disponible au sein de la distribution de l'application CTTE, dont la dernière version est disponible à l'adresse <http://giove.isti.cnr.it/tools/CTTE/>.

Le listing C.1 décrit la structure globale d'une modèle de tâches en XML. Nous avons conservé uniquement les balises pertinentes pour la construction et l'affichage

de l'arbre au sein de notre outil. La flexibilité du format XML, et son intégration aux bibliothèques JavaScript utilisées, permet de faciliter les mises à jour pour prendre en compte certains aspects non intégrés du modèle (par exemple, la coopération). Nous précisons, pour chaque élément, les attributs utilisés et leurs valeurs possibles.

Listing C.1: structure d'un modèle de tâches au format XML issu de CTTE.

```
<TaskModel>
  <Task Identifier=CDATA
    Category=(abstraction | user | interaction | application)
    Iterative=(true | false)
    Optional=(true | false)
    Frequency=CDATA >
    <Name>#PCDATA</Name>
    <Description>#PCDATA</Description>
    <TemporalOperator name=(SequentialEnabling | Disabling | Choice
      | Interleaving | Synchronization | SuspendResume |
      SequentialEnablingInfo) />
    <Parent name=CDATA />
    <SiblingLeft name=CDATA />
    <SiblingRight name=CDATA />
    <SubTask>Task*</SubTask>
  </Task>
</TaskModel>
```

Ainsi, le modèle de tâches est constitué d'une tâche « racine » respectant un schéma pour la description des tâches (balise *<Task>*). Chaque *<Task>* peut inclure un ensemble d'éléments *<SubTask>* permettant de décrire la hiérarchie de l'arbre.

Chaque élément *<Task>* comprend les attributs suivants :

- un identificateur unique (*identifier*) sous la forme d'une chaîne de caractères (CDATA), qui est également le nom affiché dans le bloc de l'application CTTE ;
- une catégorie (*category*) parmi les quatre catégories CTT définies ;
- une mention du caractère itératif (*iterative*) de la tâche sous la forme d'un booléen ;
- une mention du caractère optionnel (*optional*) de la tâche sous la forme d'un booléen ;
- une chaîne de caractères (CDATA) décrivant la fréquence de la tâche (*frequency*).

Chaque élément *<Task>* contient également les éléments suivants :

- un nom (*name*) sous la forme d'une chaîne de caractères (#PCDATA) ;
- une description (*description*) sous la forme d'une chaîne de caractères (#PCDATA) ;
- un lien vers la tâche de gauche (le cas échéant), à travers la balise *<SiblingLeft>* dont la valeur de l'attribut *name* correspond à celle de l'identificateur *Identifier* d'une tâche de même niveau hiérarchique ;
- un lien vers la tâche de droite (le cas échéant), à travers la balise *<SiblingRight>* dont la valeur de l'attribut *name* correspond à celle de l'identificateur *Identifier* d'une tâche de même niveau hiérarchique ;

- un identificateur temporel (*TemporalOperator*) dont le type est défini par l'attribut *name* de la balise `<TemporalOperator>`, défini parmi les choix d'identificateurs connus.

Notons que cet identificateur est mentionné lorsque la tâche fait partie d'une transition et qu'elle se situe à gauche du couple de tâche de la transition. Ainsi, lorsqu'un identificateur temporel est défini, la balise `<SiblingRight>` doit également contenir une référence vers la tâche suivante. Dans le cas contraire, cet identificateur temporel peut être ignoré lors de la lecture du fichier XML.

- Un élément `<SubTask>` dont le contenu correspond aux sous-tâches (qui sont des balises `<Task>` construites sur le même modèle) de la tâche décrite.

C.3.2 Description des graphes au format GraphML

Le format GraphML, initiative du groupe de travail *Graph Drawing* est décrit par le schéma DTD disponible sur le site officiel du groupe, à l'adresse <http://graphml.graphdrawing.org/specification/dtd.html>

Ce schéma très général permet aux différentes applications générant des fichiers au format GraphML d'inclure leurs propres paramètres tout en formalisant un certain nombre d'éléments afin de rendre uniforme la description structurelle du graphe. Les paramètres supplémentaires portent notamment sur l'aspect graphique et le positionnement des nœuds et arcs des graphes.

Le listing C.2 décrit la structure d'un fichier GraphML issu spécifiquement de l'application *yEd Graph Editor*. Les données structurelles du graphe sont décrites à travers un ensemble d'éléments `<node>` et `<edge>` tandis que certains paramètres propres à l'éditeur sont définis à travers les balises `<key>` et en définissant un espace de nommage préfixé par « *y:* ».

Listing C.2: structure d'un graphe GraphML exporté de l'application *yEd Graph Editor*.

```
<graph edgedefault=(directed|undirected)>
  <key for="node" id=CDATA1 yfiles.type="nodegraphics"/>
  <key for="edge" id=CDATA2 yfiles.type="edgegraphics"/>
  <node id=CDATA>
    <data key=CDATA1>
      <y:ShapeNode>
        <y:NodeLabel>#PCDATA</y:NodeLabel>
        <y:Geometry height=CDATA width=CDATA
          x=CDATA y=CDATA />
      </y:ShapeNode>
    </data>
  </node>
  <edge id=CDATA source=CDATA target=CDATA>
    <data key=CDATA2>
      <y:PolyLineEdge>
        <y:EdgeLabel x=CDATA y=CDATA>
```

```

#PCDATA
    </y:EdgeLabel>
  </y:PolyLineEdge>
</data>
</edge>
</graph>

```

Ainsi, selon le schéma DTD de GraphML, la description du graphe (balise `<graph>`) est faite par :

- un ensemble d'éléments `<node>` représentant les nœuds du graphe. Seul l'attribut d'identificateur unique *id*, décrit par une chaîne de caractères (CDATA) est nécessaire à la spécification d'un nœud..
- un ensemble d'éléments `<edge>` représentant les arêtes du graphe. Pour chaque arête, trois attributs doivent être spécifiés :
 - son identificateur unique *id* sous la forme d'une chaîne de caractères (CDATA) ;
 - l'identificateur unique du nœud source (*source*) sous la forme d'une chaîne de caractères (CDATA) ;
 - l'identificateur unique du nœud cible (*target*) sous la forme d'une chaîne de caractères (CDATA).
- Enfin, le caractère orienté ou non du graphe est spécifié par l'attribut obligatoire *edgedefault* de la balise `<graph>`.

Notons que le schéma DTD autorise l'imbrication de balises `<graph>` au sein d'autres `<graph>` afin de créer des sous-graphes. Un attribut d'identification unique *id* pourra alors être ajouté à chaque balise `<graph>` du fichier afin de différencier les sous-graphes. Dans le cadre de notre outil néanmoins, nous n'utilisons pas cette possibilité, nous considérons un graphe unique.

L'application *yEd Graph Editor* s'appuie sur l'extensibilité du schéma DTD de GraphML afin de définir un certain nombre de clés (balises `<key>`) permettant de décrire notamment le contenu et la position des nœuds et arcs. Nous ne décrivons ici que les deux clefs utilisées pour la description des éléments pertinents pour la construction et l'affichage du graphe.

Dans le DTD, il est précisé que chaque clé doit contenir les deux attributs suivants :

- un identificateur unique *id* sous la forme d'une chaîne de caractères (CDATA). Nous les avons notées CDATA1 et CDATA2 dans le listing C.2 afin de différencier les données dans ce listing.
- Le type d'élément cible (*for*), ici un nœud (valeur « *node* ») ou un arc (valeur « *edge* »).

De plus, *yEd Graph Editor* ajoute un attribut *yfiles.type* définissant le contenu de la clé :

- sa valeur est « *nodegraphics* » lorsque la clé est utilisée pour définir les données propres aux graphismes des nœuds ;
- sa valeur est « *edgegraphics* » lorsque la clé est utilisée pour définir les données

propres aux graphismes des arcs.

Ainsi, au sein de chaque élément de type `<node>` ou `<edge>`, nous retrouvons des données définies par les clés spécifiées (élément `<data>`) et dont l'attribut *key* permet de déterminer quel est le type de données contenu dans la balise.

Pour les éléments `<node>`, la balise `<data>` correspondante à la clé « *nodegraphics* » contient un élément `<y:ShapeNode>` pour lesquels sont définis :

- le libellé du nœud sous forme de chaîne de caractères (`#PCDATA`) défini par la balise `<y:NodeLabel>` ;
- la géométrie du nœud, utilisée dans notre outil pour le positionnement des nœuds. Elles sont définies par la balise `<y:Geometry>` :
 - les dimensions du nœud, définies par les attributs *height* et *width* ;
 - la position du nœud, définie par les attributs *x* et *y*.

Pour les éléments `<edge>`, la balise `<data>` correspondante à la clé « *edgegraphics* » contient un élément `<y:PolyLineEdge>` au sein duquel nous utilisons l'élément `<y:EdgeLabel>` permettant d'obtenir :

- la position relative du libellé de l'arc par rapport au nœud source, définie par les attributs *x* et *y* de la balise `<y:EdgeLabel>` ;
- le libellé de l'arc, défini par la chaîne de caractère entourée par la balise `<y:EdgeLabel>`.

L'ensemble des données géométriques ainsi récupérées lors de l'importation du fichier GraphML permettent de positionner les nœuds et arêtes en fonction des préférences de l'utilisateur. Toutes les valeurs citées étant définies dans un même référentiel propre à *yEd Graph Editor*, un changement d'origine, et éventuellement d'échelle, permet d'afficher le graphe au sein de l'outil directement à partir de ces valeurs.

C.4 DESCRIPTION DU FORMAT DES DONNÉES DE SORTIE

C.4.1 Le format JSON

Le format JSON (pour *JavaScript Object Notation*, « Notation Objet JavaScript ») est un langage de description de données qui utilise le modèle de l'objet JavaScript pour la description d'objets. Les attributs des objets sont représentés sous la forme de paires (nom, valeur), regroupées sous la forme d'un objet au sein d'accolades. La valeur d'un attribut pouvant être une chaîne de caractère, un nombre, un autre objet, une variable booléenne (VRAI/FAUX), la valeur *null* ou un tableau de valeurs parmi ces types. Ce format est entièrement spécifié à l'adresse <http://json.org/>, proposant des ressources pour la plupart des langages de programmation auxquels ce format est intégré.

En JavaScript, l'utilisation du format JSON pour la description des objets présente l'avantage de réduire le nombre d'étapes pour la manipulation des données. En effet, un modèle de données, utilisé et stocké sous la forme d'objets JSON, est

directement interprété par le moteur JavaScript et ne nécessite pas de mettre en place un mécanisme de « dissection » et de lecture (*parsing*) comme dans le cas de l'importation des fichiers XML.

Le listing C.3 décrit un objet « *Task* » tel qu'il est décrit dans le fichier XML issu de l'application CTTE.

Listing C.3: l'objet ***Task*** CTT décrit en XML

```
<Task Identifier="Task_1" Category="interaction" Iterative="false"
  Optional="false" Frequency="">
  <Name>name</Name>
  <TemporalOperator name="Choice"/>
  <Parent name="Task_0"/>
  <SiblingRight name="Task_2"/>
</Task>
```

Dans le cadre de l'application, nous utilisons ces données afin de nourrir l'attribut *cttParameters* de l'objet *Task* du modèle de données (voir aussi la section 6.3.3.2). Le listing C.4 présente la manière dont cet objet est représenté en JavaScript.

Listing C.4: la représentation en JSON des paramètres d'un objet ***Task*** issu de la lecture du fichier XML.

```
var cttParameters = {
  identifier: "Task_1",
  name: "name",
  category: "interaction",
  iterative: false,
  optional: false,
  frequency: "",
  parentName: "Task_0",
  temporalOperator: "Choice",
  siblingRight: "Task_2"
}
```

C

C.4.2 Description du projet au format JSON

Lors de la sauvegarde du projet sur le navigateur ou lorsque l'utilisateur désire exporter le fichier JSON correspondant au projet courant, nous convertissons les différentes données nécessaires à la conservation de l'état du projet pour une réutilisation future dans cet état.

Ainsi, pour chaque classe/objet JavaScript devant être exporté, une méthode *toJSON()* est définie. Celle-ci sera notamment utilisée par la bibliothèque JSON (section C.1).

Le listing C.5 décrit, au format JSON, la structure générale d'un projet. Chaque projet est composé de quatre objets de premier niveau.

- Un objet *tree*, décrit par :
 - une racine (attribut *root*) suivant le format JSON d'une tâche (listing C.6) ;
 - une liste de tâche (attribut *tasks*) suivant la même structure ;
 - une liste de transitions (attribut *transitions*), suivant le format JSON d'une transition (listing C.7) ;
 - un objet *miniaturePosition* permettant d'enregistrer les préférences d'affichage de la miniature de l'arbre sur l'interface.
- Un objet *graph*, décrit par :
 - une liste de noeuds (attribut *nodes*), suivant le format JSON d'un noeud (listing C.8) ;
 - une liste d'arêtes (attribut *edges*), suivant le format JSON d'une arête (listing C.9) ;
 - un objet *miniaturePosition*.
- Une liste d'analyses, suivant le modèle d'objet d'une transition (listing C.7).
- Une chaîne de caractères définissant le nom du projet (attribut *projectName*).

Listing C.5: structure du format JSON définissant un projet dans l'outil SAMANTA.

```

var project = {
  tree: {
    root: Task,
    tasks: [Task, Task...],
    transitions: [Transition, Transition, ...],
    miniaturePosition: {
      left: int,
      top: int
    }
  },
  graph: {
    nodes: [Node, Node, ...],
    edges: [Edge, Edge, ...],
    miniaturePosition: {
      left: int,
      top: int
    }
  },
  analyses: [Transition, Transition, ...],
  projectName: String
}

```

Le format JSON pour la description d'une tâche de l'arbre est donné par le listing C.6. Chaque tâche est identifiée par un entier unique, permettant de référencer les tâches entre elles. Pour chaque tâche, un attribut *context* permet de stocker les différentes données relatives au contexte défini pour cette tâche (liste des noeuds et des noeuds saillants pour ce contexte).

La liste des tags est stockée sous la forme d'un tableau associatif où les clés sont les identificateurs uniques des noeuds (une chaîne de caractère unique) et les valeurs des listes de chaînes de caractère définissant les tags associés à ce noeud.

Listing C.6: structure de la description JSON d'une tâche CTT.

```

Task: {
  id: int ,
  pid: int ,
  dummy: boolean ,
  leftSiblingId: int ,
  rightSiblingId: int ,
  ctt:{
    name: String ,
    type: String ,
    description: String ,
    identifier: String ,
    category: String ,
    iterative: boolean ,
    optional: boolean ,
    partOfCooperation: boolean ,
    frequency: String ,
    platform: String ,
    precondition: String ,
    timePerformance: String
  },
  context: {
    tags: [
      (String)"idNode1 ":[ String ,... ] ,
      (String)"idNode2 ":[ String ,... ] ,
      ...
    ] ,
    nodeIds:[ String ,... ] ,
    salientNodeIds:[ String ,... ]
  },
  childrenId:[ int ,... ]
}

```

Le format JSON pour la description d'une transition est présenté dans le listing C.7. Celui-ci contient les références, sous forme d'identificateur unique, aux deux tâches concernées, à l'opérateur les reliant. Un booléen *fake* permet de définir si la transition est une « vraie » transition de l'arbre CTT (liant deux tâches adjacentes) ou s'il s'agit d'une transition définie par l'utilisateur lors de l'analyse.

Listing C.7: structure de la description JSON d'une transition CTT.

```

Transition: {
  leftId: int ,
  rightId: int ,
  operator: String ,
  fake: boolean
}

```

Le format JSON pour la description d'un nœud du graphe est présenté dans le listing C.8. Notons que l'identificateur unique *id* est une chaîne de caractère, issue des données XML. De la même façon, la position et la taille du nœud sont sauvegardés, car il s'agit de données d'entrée définies lors de l'importation du fichier

GraphML décrivant ce graphe.

Listing C.8: structure de la description JSON d'un nœud du graphe.

```
Node: {  
  id: String,  
  xPosition: int ,  
  yPosition: int ,  
  width: int ,  
  height: int ,  
  label: String  
}
```

Le format JSON pour la description d'une arête du graphe est présenté dans le listing C.9. Notons que la position relative du libellé (par rapport au nœud source) est enregistrée à travers les attributs (*xShift*, *yShift*) de l'objet *labelPos*. Ces données sont issues du placement des libellés par l'utilisateur lors de la création du graphe, au sein de l'application *yEd Graph Editor* et sont des données d'entrée définies lors de l'importation du fichier GraphML décrivant ce graphe.

Listing C.9: structure de la description JSON d'une arête du graphe.

```
Edge: {  
  label: String,  
  id: String,  
  sourceId: String,  
  targetId: String,  
  labelPos: {  
    xShift: int ,  
    yShift: int  
  }  
}
```



Le PEA FH/PA

Afin d'appliquer notre méthodologie aux systèmes de drones, nous nous appuyons sur les données collectées et les résultats du Programme d'Étude Amont (PEA) « Étude des Facteurs Humains (FH) et Partage d'Autorité (PA) homme/système dans les systèmes de drones » (FH/PA) financé par la Direction Générale de l'Armement (DGA).

D.1 ORGANISATION DU PROJET

Quatre entreprises ont collaboré sur ce projet :

Bertin Technologies, entreprise mandataire du groupement était en charge du management du projet. Elle apportait également son expertise en ergonomie et en facteurs humains sur les différentes phases de l'étude.

Cassidian Air Systems (groupe EADS) et **SAGEM Defense & Security (groupe Safran)** apportaient leur expertise sur les systèmes de drones MALE (Moyenne Altitude Longue Endurance) et tactiques, présentés dans le paragraphe 7.1.

PY Automation apportait son expertise en algorithmie et en génie logiciel, notamment dans la conception et réalisation du logiciel de simulation S4D utilisé pour l'évaluation des solutions proposées dans le cadre de l'étude.

D.2 OBJECTIFS ET APPROCHE

Ce projet, terminé au premier semestre 2011, avait pour objectif la définition et la validation d'une méthode de conception appliquée aux systèmes de drones pour rendre opérationnel le concept de Partage d'Autorité. Ce concept, défini comme une « stratégie d'utilisation des ressources des agents (humain et/ou automate) du système considéré », se traduit par « la capacité d'un tel système d'allouer, à chaque instant et en fonction du contexte, une tâche précise à l'agent le plus à même de l'effectuer dans les conditions optimales exigées d'efficacité (atteinte des objectifs de la tâche) et de sécurité (intégrité du matériel, des personnes et de l'environnement) » (Leal *et al.*, 2009).

L'une des finalités de l'étude est l'optimisation de la relation $\langle p, n, q \rangle$ (p opérateurs contrôlant n drones dans q situations opérationnelles). Pour atteindre ce but,

le groupement d'entreprises a pris le parti d'étudier des configurations inexistantes où deux opérateurs (un opérateur Vecteur et un opérateur Charge Utile) sont en charge de deux drones simultanément.

Au cours de cette étude, une approche centrée sur les utilisateurs a été adoptée afin d'intégrer à la méthode de conception les différents besoins, exigences et limites issus des opérationnels des Armées, des métiers des drones (contacts industriels) et du domaine des facteurs humains (exigences pour l'ergonomie des systèmes).

D.3 RÉSULTATS THÉORIQUES

La méthodologie de conception issue de cette étude se déroule en quatre étapes (Leal *et al.*, 2009) :

1. analyse et modélisation des macro-tâches des opérateurs du système considéré ;
2. définition d'un ensemble de modes opératoires déterminant les différentes configurations possibles, en termes d'allocation des tâches entre l'humain et l'automate, pour la réalisation des macro-tâches considérées ;
3. descriptions détaillées des macro-tâches en intégrant les modes opératoires définis dans l'étape 2 ;
4. maquetage des IHM.

Cette méthodologie a été appliquée à l'étude et la conception d'interface pour la macro-tâche de « gestion de la trajectoire » de l'opérateur Vecteur (Villaren *et al.*, 2010). Elle a été évaluée à travers deux campagnes d'expérimentations. Au cours de ces campagnes, des opérationnels des Armées et des experts drones issus de l'industrie ont mené à bien des scénarios opérationnels sur le simulateur S4D où avaient été intégrées les solutions de l'étude, dont les IHM conçues afin de soutenir cette macro-tâche.

D.4 RÉSULTATS DES ÉVALUATIONS

Lors des expérimentations, deux stations drones étaient simulées en parallèle : une station tactique (type Sperwer) et une station MALE (type Harfang). Dans la station tactique, un binôme d'opérateur (opérateur Vecteur et opérateur Charge Utile) contrôlait deux drones tactiques simultanément. La station MALE, composée également d'un binôme d'opérateurs, ne contrôlait qu'un drone, mais au cours du scénario, était amenée à contrôler deux charges utiles simultanément.

Afin de soutenir la macro-tâche de gestion de la trajectoire du drone, un mode proposant une allocation des tâches intermédiaire aux modes « classiques » manuel et automatique a été mis en place (Villaren *et al.*, 2010) et évalué. Les moyens d'interaction proposés à travers ce mode ont été très appréciés des opérateurs, car

il facilitait les interactions par rapport à un mode manuel, tout en offrant un degré de liberté et des informations supplémentaires inexistantes en mode automatique.

Les deux séries d'expérimentations ont également permis de mettre en exergue les résultats suivants :

- Le pilotage de deux vecteurs en parallèle est possible *dans le cadre de la simulation* même si certaines situations impliquant une forte charge de travail de l'opérateur vecteur rendaient la tâche plus difficile à gérer. Extrapolé en opération extérieur, le pilotage de deux drones simultanément ne serait possible qu'avec l'appui d'automatismes robustes et envers lesquels l'opérateur aura une confiance totale.
- De façon beaucoup plus franche, la charge de travail induite lors du contrôle de deux charges utiles simultanément et sur deux objectifs de missions distincts est trop importante pour pouvoir mener à bien les deux missions. En effet, il s'est avéré très difficile de passer d'une Charge Utile à l'autre lorsque le contexte autour du point visé n'est pas le même, chacune des missions impliquant une attention importante pour être menée à bien.



Compléments sur l'application aux systèmes de drones

Dans cette annexe, nous présentons les contextes définis dans le cadre de la mise en oeuvre de la méthodologie SAMANTA pour la conception d'un système de contrôle/commande de drone (chapitre 7). Seuls les contextes utiles à l'analyse des deux transitions étudiées dans la section 7.6 sont décrits.

Par soucis de concision, nous affichons à chaque fois une miniature du contexte associé ainsi qu'une liste textuelle des éléments associés. Nous invitons le lecteur à se référer aux figures de la section 7.4.2 pour retrouver l'emplacement précis de chaque éléments.

E.1 SOUS-TÂCHES DE « GÉRER L'ENVIRONNEMENT »

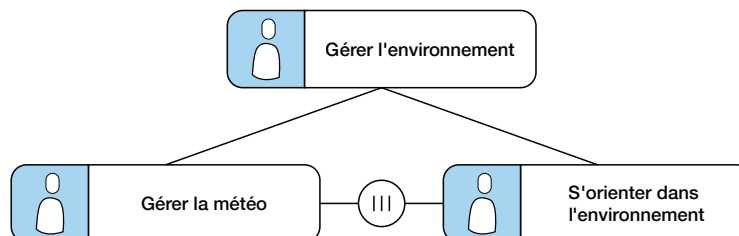


FIGURE E.1 : la tâche « Gérer l'environnement » et ses sous-tâches.

La tâche « Gérer l'environnement » étant exécutée en parallèle des transitions étudiées, les contextes qui lui sont associés peuvent être pris en compte lors de l'analyse. Cette tâche est composée de deux sous-tâches : « Gérer la météo » et « S'orienter dans l'environnement ».

E.1.1 Contexte associé à la tâche « Gérer la météo »

Les éléments associés à la tâche « Gérer la météo » sont *éléments météorologiques, nuages, vent, bruit, direction du vent, vitesse du vent, température et givre*. Ils sont représentés sur la figure E.2.



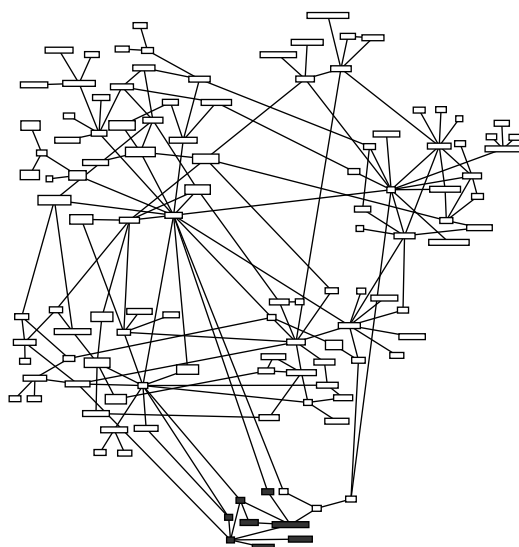


FIGURE E.2 : contexte associé à la tâche « Gérer la météo ».

E.1.2 Contexte associé à la tâche « S'orienter dans l'environnement »

Les éléments associés à la tâche « S'orienter dans l'environnement » sont *relief, bâtiment, carrefour, point de repère, route, champ, zone, entité cartographique, plan de vol, point de passage, point cardinal, Nord, cartographie, repère "aquatique", Point d'Intérêt, forme géométrique et hauteur du bâtiment*. Ils sont représentés sur la figure E.3.

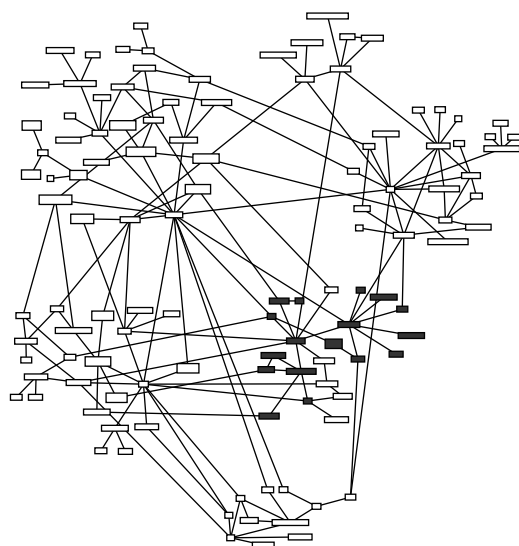


FIGURE E.3 : contexte associé à la tâche « S'orienter dans l'environnement ».

E.2 SOUS-TÂCHES DE « PILOTER LE DRONE »

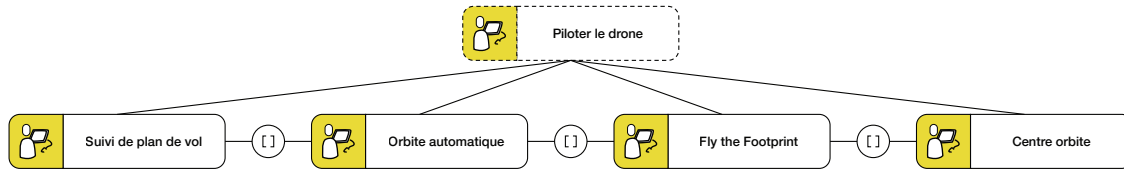


FIGURE E.4 : la tâche « Piloter le drone » et ses sous-tâches.

La tâche « Piloter le drone » étant exécutée en parallèle des transitions étudiées, les contextes qui lui sont associés peuvent être pris en compte lors de l'analyse. Cette tâche est composée de quatre sous-tâches : « Suivi de plan de vol », « Orbite automatique », « Fly The Footprint » et « Centre orbite ».

E.2.1 Contexte associé à la tâche « Suivi de plan de vol »

Les éléments associés à la tâche « Suivi de plan de vol » sont *relief, plan de vol, point de passage, bruit, température, givre, black hot, trajectoire verticale, montée, descente, vitesse du drone, drone, mode suivi de plan de vol, mode de contrôle du drone et position du drone*. Ils sont représentés sur la figure E.5.

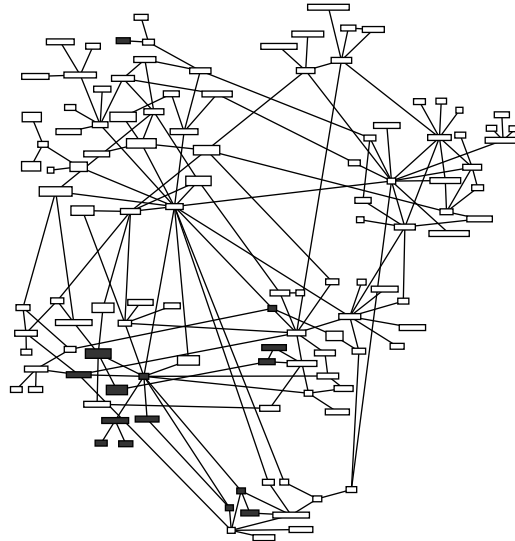


FIGURE E.5 : contexte associé à la tâche « Suivi de plan de vol ».

E.2.2 Contexte associé à la tâche « Orbite automatique »

Les éléments associés à la tâche « S'orienter dans l'environnement » sont *zone, type de zone, forme de la zone, zone restreinte, volume spatial, vent, bruit, direc-*

tion du vent, vitesse du vent, inclinaison de la charge utile (*pitch*), trace senseur, trajectoire verticale, montée, descente, vitesse du drone, distance, drone, partie du drone (*train, ailes...*), mode "orbite automatique", mode de contrôle du drone, azimut, position du drone, élévation et position relative. Ils sont représentés sur la figure E.6.

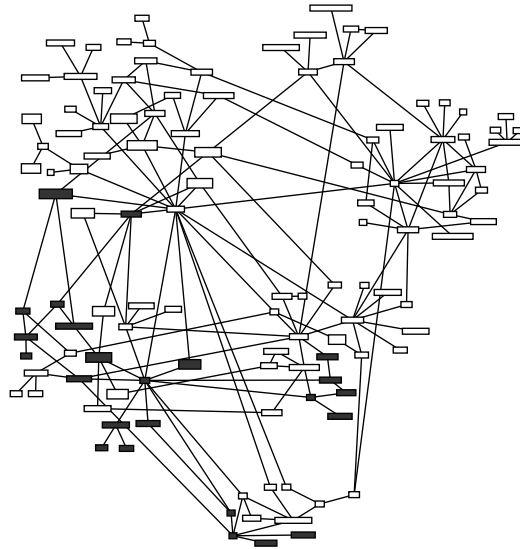


FIGURE E.6 : contexte associé à la tâche « Orbite Automatique ».

E

E.2.3 Contexte associé à la tâche « Fly The Footprint »

Les éléments associés à la tâche « Fly The Footprint » sont *zone, type de zone, forme de la zone, zone restreinte, volume spatial, vent, bruit, direction du vent, vitesse du vent, charge utile, trace senseur, trajectoire verticale, montée, descente, vitesse du drone, drone, mode Fly The Footprint (FTF), mode de contrôle du drone, position absolue, altitude et position du drone*. Ils sont représentés sur la figure E.7.

E.2.4 Contexte associé à la tâche « Centre orbite »

Les éléments associés à la tâche « Centre orbite » sont *zone, type de zone, forme de la zone, zone restreinte, volume spatial, Point d'Intérêt, vent, bruit, direction du vent, vitesse du vent, trajectoire verticale, montée, descente, vitesse du drone, drone, mode centre orbite, mode de contrôle du drone, position absolue, altitude, position du drone, latitude et longitude*. Ils sont représentés sur la figure E.8.

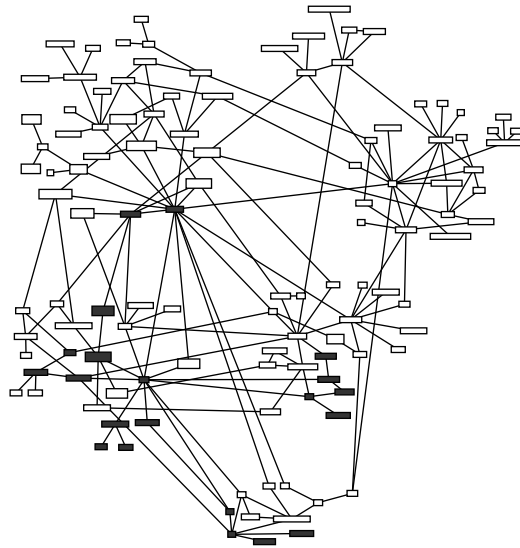


FIGURE E.7 : contexte associé à la tâche « Fly The Footprint ».

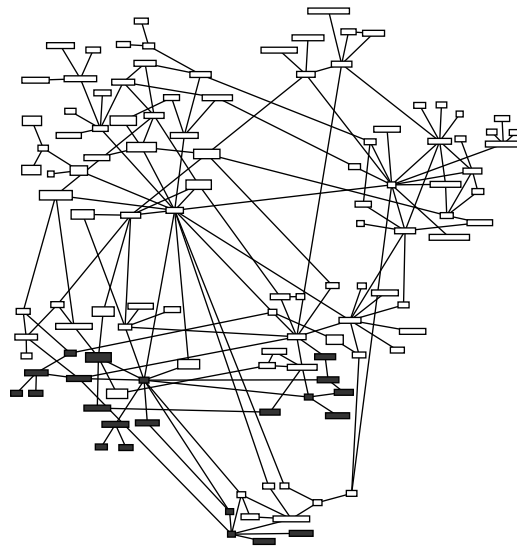


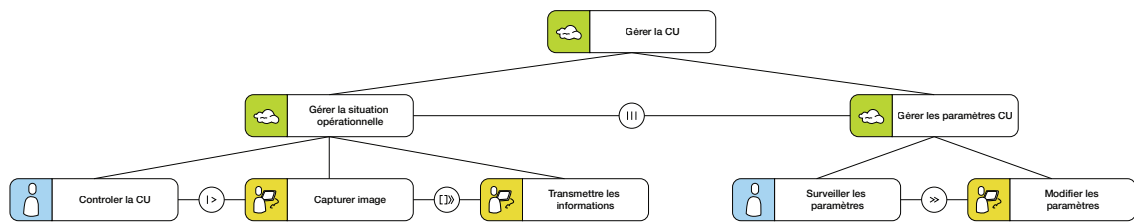
FIGURE E.8 : contexte associé à la tâche « Centre orbite ».

E.3 SOUS-TÂCHES DE « GÉRER LA CHARGE UTILE »

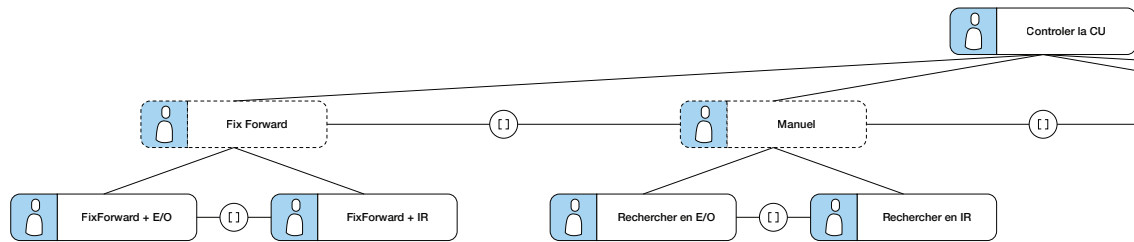
En parallèle des transitions étudiées peut être exécutée la sous-tâche « Gérer les paramètres CU ».

Les modes de contrôle de la Charge Utile exécutés lors du déroulement du scénario présenté dans la section 7.6.1 sont les tâches « FixForward + E/O », « Rechercher en E/O », « Rechercher en IR » et « Surveiller cible en IR ».

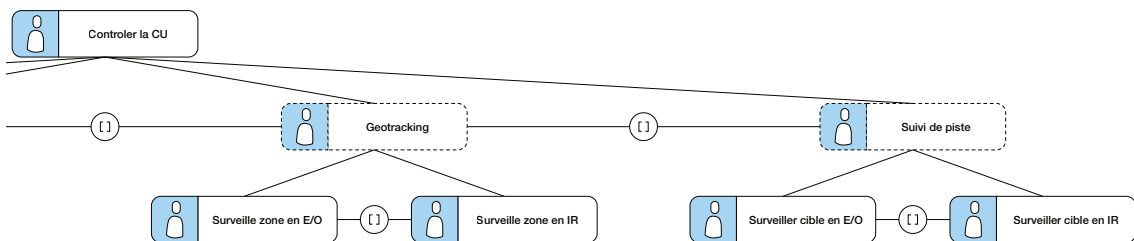
Nous présentons les contextes associés à ces cinq tâches dans cette section.



(a) sous-tâches directes de la tâche « Gérer la Charge Utile ».



(b) sous-tâches de la tâche « Contrôler la Charge Utile » (partie gauche).



(c) sous-tâches de la tâche « Contrôler la Charge Utile » (partie droite).

FIGURE E.9 : tâche « Gérer la Charge Utile » et ses sous-tâches.

E

E.3.1 Contexte associé à la tâche « Gérer les paramètres CU »

Les éléments associés à la tâche « Gérer les paramètres CU » sont *type de charge utile, polarité, palier de zoom, ouverture, panne, flux vidéo de la caméra, stabilisation de la caméra, paramètre de contraste, paramètre, statut de la charge utile, netteté, vitesse d'obturation, niveau de zoom, mode de contrôle de la Charge Utile* et *OK*. Ils sont représentés sur la figure E.10.

E.3.2 Contexte associé à la tâche « FixForward + E/O »

Les éléments associés à la tâche « FixForward + E/O » sont *éléments météorologiques, nuages, soleil, type de charge utile, Electro/Optique (E/O), panne, flux vidéo de la caméra, mode de contrôle de la Charge Utile* et *mode Fix Forward*. Ils sont représentés sur la figure E.11.

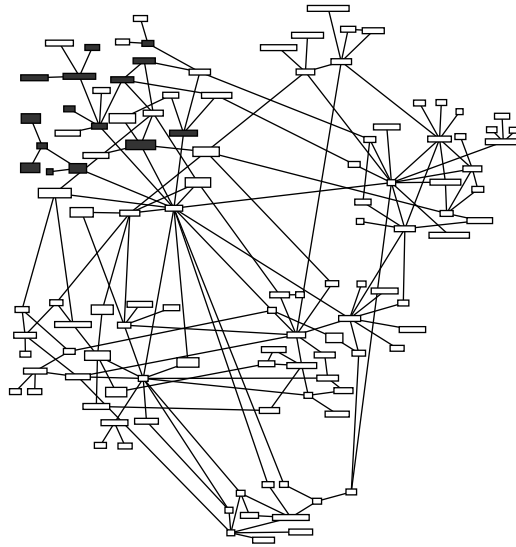


FIGURE E.10 : contexte associé à la tâche « Gérer les paramètres CU ».

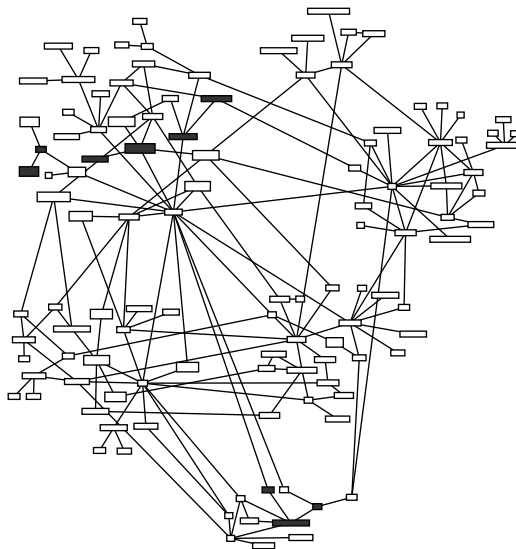


FIGURE E.11 : contexte associé à la tâche « FixForward + E/O ».

E.3.3 Contexte associé à la tâche « Rechercher en E/O »

Les éléments associés à la tâche « Recherche en E/O » sont *bâtiment, carrefour, point de repère, route, champ, forme géométrique, ombre, nuages, soleil, reflet, type de charge utile, Electro/Optique (E/O), netteté, niveau de zoom, mode de contrôle de la Charge Utile, inclinaison de la charge utile (pitch), orientation de la charge utile (yaw), trace capteur, mode manuel, historique de la trace capteur, portée DRI, piste, orientation de la piste (visée), type de piste, comportement, position de la piste, ennemi, ami, neutre, statut de la piste, dynamique de la piste, vitesse de la piste,*

arme à feu, humain, animal, véhicule, aérien, environnement de la piste, terrestre, aquatique et couleur. Ils sont représentés sur la figure E.12.

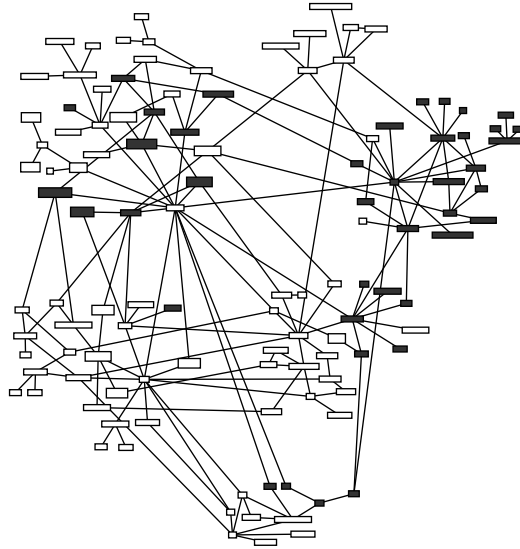


FIGURE E.12 : contexte associé à la tâche « Rechercher en E/O ».

E

E.3.4 Contexte associé à la tâche « Rechercher en IR »

Les éléments associés à la tâche « Rechercher en IR » sont *nuages, type de charge utile, polarité, Infrarouge (IR), palier de zoom, white hot, black hot, niveau de contraste, paramètre de contraste, mode de contrôle de la Charge Utile, inclinaison de la charge utile (pitch), orientation de la charge utile (yaw), trace senseur, mode manuel, historique de la trace senseur, portée DRI, piste, orientation de la piste (visée), type de piste, comportement, position de la piste, ennemi, ami, neutre, statut de la piste, dynamique de la piste, vitesse de la piste, arme à feu, humain, animal, véhicule, aérien, environnement de la piste, terrestre, aquatique et chaleur.* Ils sont représentés sur la figure E.13.

E.3.5 Contexte associé à la tâche « Surveiller cible en IR »

Les éléments associés à la tâche « Surveiller cible en IR » sont *relief, bâtiment, carrefour, point de repère, route, hauteur du bâtiment, nuages, type de charge utile, polarité, Infrarouge (IR), palier de zoom, white hot, black hot, stabilisation de la caméra, niveau de contraste, paramètre de contraste, mode de contrôle de la Charge*

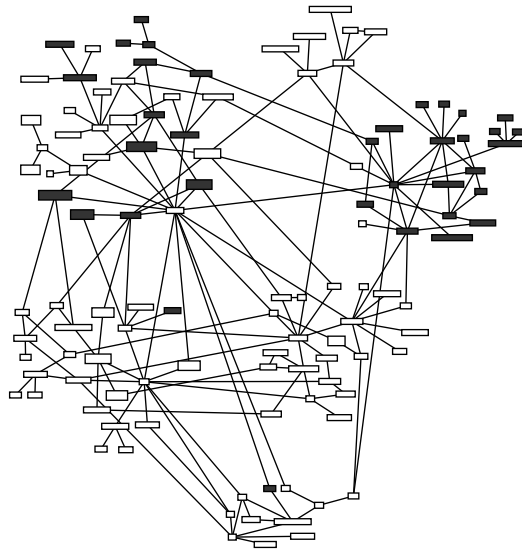


FIGURE E.13 : contexte associé à la tâche « Rechercher en IR ».

Utile, mode verrouillage géolocalisé, partie du drone (train, ailes...), piste, orientation de la piste (visée), type de piste, comportement, position de la piste, ennemi, ami, neutre, statut de la piste, dynamique de la piste, vitesse de la piste, arme à feu, humain, véhicule, aérien, environnement de la piste, terrestre, aquatique et chaleur. Ils sont représentés sur la figure E.14.

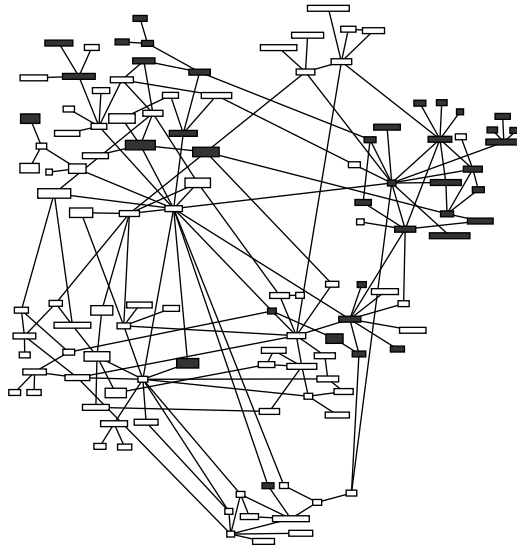


FIGURE E.14 : contexte associé à la tâche « Surveiller cible en IR ».

Résumé

Les systèmes complexes actuels (cockpits, contrôle de trafic aérien, pilotage de drones, ...) amènent les opérateurs à gérer un ensemble de tâches évoluant dynamiquement dans le temps, et imposent des changements fréquents de situation au cours d'une même mission. Ces transitions entre contextes (provoqués par l'ajout, le retrait, l'interruption ou la récupération d'une tâche) peuvent induire une rupture de la conscience de situation (Situation Awareness) de l'opérateur.

Dans le cadre de cette thèse, nous introduisons la méthodologie SAMANTA (Situation Awareness Modeling and ANalysis for Transition Amelioration). Celle-ci, mise en oeuvre par les experts en charge de l'analyse et de la conception de futurs systèmes, contribue à l'étude et la résolution des difficultés engendrées par de telles transitions. Le processus d'analyse des transitions proposé s'appuie sur une approche duale de description des transitions à travers la modélisation des tâches, et des éléments de situation associés à chacune d'entre elles (les contextes). Les caractéristiques des transitions ainsi obtenues sont utilisées par l'analyste afin d'émettre des recommandations sur les interfaces du système considéré. Les utilisateurs (opérateurs de systèmes et experts du domaine) sont intégrés aux étapes clés du processus de conception.

Un outil dédié a été développé dans le cadre de cette thèse dans le but de faciliter la mise en oeuvre de la méthodologie SAMANTA par les analystes. Une application concrète à la conception de futurs systèmes de drone est présentée, développant l'étude de deux transitions spécifiques.

Mots clés : Interaction homme-machine, Changement de tâches, Méthodologie de conception, Facteurs Humains, Conscience de la situation, Modélisation, Interruption.

Abstract

Complex systems such as aircraft cockpits, Air Traffic Control consoles or Unmanned Vehicle control stations, operators manage a set of tasks with high temporal dynamics and inducing frequent changes of situation within the same mission. These transitions between contexts (triggered by the addition, removal, interruption or retrieval of a task) may reduce operator's Situation Awareness.

In this thesis, we introduce SAMANTA (Situation Awareness Modeling and ANalysis for Transition Amelioration), a model-based and user-centered methodology. Designed for Human Factors experts in charge of future systems' analysis and design, this methodology helps identifying and solving the difficulties caused by such transitions. The transitions analysis process is based on the dual description of operators' activity: the modeling of operator's task is associated with the definition of Situation Awareness requirements for each of their tasks (the tasks' context). The experts base their analysis on the characteristics obtained and provide recommendations for the new interfaces in order to support the foreseen impacts. End-users and Subject Matter Experts are involved at different key stages of the design process supported by this methodology.

In order to support the implementation of SAMANTA by the analysts, we developed a specific web-based tool. The methodology has been applied to the design of future Unmanned Aerial Vehicle systems, detailing the analysis of two specific transitions.

Keywords : Human-computer Interaction, Task transitions, Design methodology, Human factors, Situation Awareness, Models, interruption.